Universidad Autónoma Metropolitana

CBI

Tres heurísticas basadas en inteligencia de partículas adaptadas al problema de asignación generalizada

Tesis para obtener el grado de Maestro en Optimización

presenta:

Gilberto Sinuhé Torres Cockrell

Asesores:
Dr.Javier Ramírez Rodríguez
Dr.Roman Anselmo Mora Gutiérrez

Dedicado a
mi familia

"Un emprendedor
ve oportunidades
allá donde otros
solo ven problema".
Michael Gerber

Es preferible obtener
una respuesta razonablemente aproximada
pero rápida que le indique si el diseño funciona o no,
que invertir más tiempo y obtener el mismo resultado
sólo que con más decimales.
Robert L Norton(Libro Diseño de máquinas)

El camino hacia el éxito
siempre esta en construcción.
Lily Tomlin

Agradecimientos

Gracias principalmente a ***CONACYT*** que me brindó los medios para poder hacer esta etapa de mi vida posible, dándome la oportunidad de realizar investigación, salir a eventos llenos de conocimiento y conocer especialistas del medio que me aportaron críticas constructivas e ideas interesantes, ya que sin este gran apoyo no hubieran sido posibles. Gracias a DIOS por guiar siempre de su mano mi camino en la vida, rodeándome de las personas idóneas que me han brindado su ayuda y sus conocimientos, por darme unos padres ejemplares y amorosos que fueron los primeros en prepararme para afrontar todos los retos de la vida con amor, devoción y sacrificios. Brindándome la oportunidad de salir adelante y mostrándome que el conocimiento es la llave que abre muchas puertas, motivo por el cual les estaré agradecido el resto de mis días, pues nunca podré pagarles todo lo que han hecho por mí, gracias a mis hermanos que me enseñaron de niño a ser un ejemplo a seguir para que ellos siempre trataran de ser mejores y por apoyarme en los momentos difíciles, a mi esposa por apoyarme siempre en las buenas y en las malas en esta etapa de mi vida, sacrificando tiempo de convivencia para que logre alcanzar mis metas, no tengo palabras para agradecerle su paciencia y apoyo incondicional, a mis asesores por brindarme su amistad, sus conocimientos, su tiempo y desvelos, apoyándome siempre y motivándome a ser mejor cada día, me enseñaron que siempre hay oportunidades cuando uno tiene ganas de hacer las cosas, gracias por su amistad y por toda la comprehensión brindada, son un ejemplo a seguir, agradezco también a mis amigos que me acompañaron en este camino lleno de conocimiento haciéndolo más ligero y agradable, por estar ahí cuando los necesité como un gran equipo y por último y no menos meritorio agradezco a los profesores que nos brindaron sus conocimientos, apoyo y críticas durante nuestra estancia en la maestría.

Índice general

Índice de figuras

Índice de cuadros

Resumen

El presente trabajo desarrolla la adaptación de tres técnicas heurísticas pertenecientes a la rama de las metaheurísticas denominada inteligencia de partículas (IP) para su adaptación al problema de asignación generalizada (PAG).

Las heurísticas adaptadas al problema de asignación generalizada son las siguientes:

- Algoritmo de luciérnagas (AL)
- Búsqueda Gravitacional (ABG)
- Método de composición musical (MCM)

Se propone una codificación para representar una solución, que permita visualizar de una manera fácil las características de la misma. A diferencia de los algoritmos originales que comienzan con soluciones aleatorias, aquí se plantea una estrategia para generar soluciones de buena calidad aprovechando las características del problema lo cual genera una adaptación en los métodos antes mencionados, ya que se agregan dos parámetros los cuales permitirán generar soluciones por tres estrategias diferentes los cuales serán representados de la siguiente manera: P_g (% soluciones realizadas por un algoritmo glotón) y P_a (% de soluciones realizadas de manera aleatoria) el restante de la población será generada por una relajación del problema con fijación de variables y generación aleatoria.

Para la estrategia AL se realizaron las siguientes adaptaciones: la función objetivo no solo es representada por el valor del individuo sino que en base a su aptitud a través de las reglas de manejo de restricciones de Coello 2002 [14]. La distancia entre dos individuos no es determinada solamente por la distancia euclidiana, sino que también contempla la distancia de Hamming la cual permite encontrar soluciones de buena calidad; se generaron seis versiones del algoritmo de luciérnagas las cuales se basan en 4 vecindarios diferentes.

Para la estrategia ABG se realizaron las siguientes adaptaciones: la masa de inercia es calculada por las reglas de manejo de restricciones de Coello 2002 la cual genera una función de pesos; la función de distancia propuesta contempla al igual que en el AL la distancia euclidiana y la distancia de Hamming, para este algoritmo tanto la mejor como la peor solución el movimiento lo realiza con una búsqueda local; para el resto de las soluciones el movimiento lo realiza con base a las características de la estrategia orinal.

Para la estrategia MCM se realizaron las siguientes adaptaciones: la función objetivo utiliza las reglas de manejo de restricciones de Coello 2002 para determinar la aptitud de la solución; la función de distancia propuesta contempla al igual que en las estrategias AL y ABG la distancia euclidiana y la distancia de Hamming.

Se tomaron instancias de prueba utilizadas en Chu y Beasley 1997[13], las cuales se encuentran disponibles en Or-Library.

Se realizó una calibración de parámetros para las estrategias adaptadas al PAG con la estrategia de búsqueda armónica (BA) para cien mil llamadas a la función objetivo.

Se llevó a cabo una serie de experimentos y una comparación con base en algunas de las estrategias del estado del arte del problema.

Los resultados obtenidos permiten afirmar que se obtienen soluciones de buena calidad.

Capítulo 1

Introducción

En este trabajo se muestran las propuestas de adaptación de tres técnicas heurísticas poblacionales para resolver el problema de asignación generalizada *(PAG)*.

De manera simple se puede definir el *PAG* como un **problema combinatorio**, en el cual se trata de dividir un conjunto de "n" elementos en "m" grupos con capacidad limitada a costo minimo; el cual fué introducido por De Maio y Roveda en 1971 [18] donde describen un problema especial de transporte y la formulación coincide con uno de los problemas conocidos como problemas de Hitchcock , pero es hasta 1975 cuando es nombrado por Ross y Soland como *PAG* [61], este problema ha sido resuelto por diversas técnicas tanto exactas como héuristicas; entre las exactas se puede mencionar ramificación y acotamiento, relajación lagrangiana, ramificación y corte, por otro lado en el caso de técnicas heurísticas se pueden mencionar a: búsqueda tabú, algoritmos genéticos, entre otros.

Se hace notar, que tanto el concepto como una revisión profunda del estado del arte son exhibidos con mayor detalle en el capítulo 2.

En los últimos cincuenta años, el *PAG* ha sido de gran interés para la comunidad científica e industrial ya que este se encuentra inmerso en otros problemas industriales de gran interés económico tales como: ruteo de vehículos *(VRP)*, producción, calendarización, localización, entre otros.

Una familia de metaheurísticas denominada inteligencia de partículas *(IP)* han tomado fuerza en los los últimos veinte años para resolver problemas como los antes mencionados; las características de dicha familia consiste en imitar el comportamiento de sistemas biológicos y tiene la particularidad de

ser del tipo poblacional, este tipo de metaheurísticas han presentado excelente comportamiento para problemas en espacio continuo lo que ha despertado el interés de la comunidad a realizar discretizaciones de estas.

Se observó un nicho de oportunidad en algoritmos como: luciérnagas *(AL)* y de búsqueda gravitacional *(ABG)*. Puesto que; en una revisión del estado del arte se encontró que han sido adaptadas al *VRP* obteniendo buenos resultados. Debido a que el *PAG* como se mencionó con anterioridad es un subproblema del *VRP* despertó un interés científico en proponer sus adaptación para el *PAG*. Puesto que, para este problema en particular en la revisión del estado del arte aún no han sido implementadas. Así mismo, siguiendo el camino las metaheurísticas pertenecientes a la *IP* se encontró un método mas reciente a los anteriores llamado: método de composición musical *(MCM)*; que a pesar de que no ha sido implementado al *VRP* ha sido aplicado a problemas discretos obteniendo resultados prometedores.

Para la implementación de las tres técnicas heurísticas antes mencionadas en su proceso de adaptación al *PAG* se desarrollaron seis versiones para el *AL*, una para el *ABG* y una para el *MCM*, durante este proceso se realizaron algunas modificaciones a los algoritmos originales; una de ellas fue que los algoritmos no parten de una población de soluciones aleatorias. Sino que, la población inicial es construida en tres bloques; el primer bloque es obtenido por un generador de soluciones glotonas, el segundo bloque es producido por un generador de soluciones con base a una relajación del problema y para no perder diversidad el tercer bloque esta compuesto por soluciones aleatorias, lo que generó la incorporación de dos parámetros a las técnicas antes mencionadas, el resto de las modificaciones serán mencionadas en el capitulo 4.

Las discretizaciónes obtenidas de las técnicas desarrolladas fueron comparadas con algunas técnicas reportadas en la literatura, en cuestión de calidad de la solución; para este trabajo se utilizaron las instancias de P. C. Chu y J. E. Beasley propuestos en [13], los archivos se encuentran disponibles en **OR-Library** [10].

En este trabajo, los resultados obtenidos por las discretizaciones mostraron tener comportamientos estables y son de orden ("N^2"), donde las mejores soluciones son obtenidas por los algoritmos *AL* y *ABG*; que demuestran ser estadísticamentes similares, mientras que el *MCM* muestra el peor comportamiento de las implementaciones desarrolladas y es estadísticamente diferente a las dos técnicas antes mencionadas. Dichas técnicas implementadas mostraron tener resultados prometedores para cien mil llamadas a la función objetivo a pesar de que los algoritmos no convergen a las las mejores solucio-

nes existentes hasta el momento brindan una ventana de oportunidad para realizar como trabajo a futuro hibridaciones con uno delos algoritmos que logra obtener las mejores soluciones pero que resultan ser mas robustos.

1.1. Objetivo General

Adaptar los métodos heurísticos de luciérnagas *AL*, gravitación universal *ABG*, composición musical *MCM* para resolver algunas instancias del problema de asignación generalizada.

1.2. Objetivos Particulares

- Desarrollar al menos una estrategia para generar soluciones iniciales de buena calidad para el conjunto de instancias propuestas en [13] y disponibles en [10].
- Adaptar, implementar y describir el comportamiento de los métodos propuestos *(AL, ABG y MMC)* para el *PAG* con las instancias seleccionadas.
- Comparar los resultados obtenidos de las tres heurísticas con algunos resultados obtenidos en la literatura.
- Identificar las ventajas y desventajas de los algoritmos implementados al *PAG*.

Capítulo 2

Conceptos y marco de referencia

En este trabajo, se proponen algunos métodos heurísticos para encontrar soluciones al problema de asignación generalizada *PAG* del tipo minimizar; por lo que en el presente capítulo se enuncian los conjunto de conceptos implicados en el mismo.

En términos comunes, el *PAG* es un **problema combinatorio**, en el cual se trata de dividir un conjunto de "n" elementos en "m" grupos con capacidad limitada.

Como se mencionó, el *PAG* está compuestos por dos conjuntos, uno de "n" tareas y otro de "m" agentes. De acuerdo a la Real Academia Española (RAE),*una tarea se encuentra definida como una obra o trabajo que debe hacerse en un tiempo limitado, que conlleva una penalidad o cuidado causado por un trabajo continuo*; por otro lado, la *(RAE)* define a un agente *como una persona, animal o cosa que tiene capacidad de obrar produciendo un efecto o acción.*

Particiona el *PAG* un conjunto de "n" tareas en "m" conjuntos cada uno de los cuales es asignado a uno de los m agentes. Cabe mencionar que el conjunto de agentes es heterogéneo por ende cada agente *"j"* tiene una capacidad a_{ij} para realizar las tareas que se le asignen, cuya suma de capacidades no debe exceder la capacidad del agente; ya que cada una de las tareas que realiza disminuye su capacidad. Además, todas la tareas deben ser asignadas y una condición importante de este problema es que todas las tareas deben ser asignadas solo un vez, de tal manera que si un agente la realiza ningún otro podrá tomarla. Esta combinación de conjuntos tienen un impacto económico

que depende de las características del agente que la realiza.

El *PAG* fue modelado como un problema de programación matemática del tipo ***programa lineal entero***, se debe hacer notar que el *PAG* puede modelar varios problemas reales, tales como: ruteo de vehículos, localización, cadenas de suministro, calendarización, entre otros.

El PAG se puede representar con el siguiente programa lineal entero.

Los parámetros se definen como:
$i =$ conjunto de tareas $(i = 1, ..., n)$.
$j =$ conjunto de agentes $(j = 1, ..., m)$.
$b_j =$ capacidad del agente j.
$a_{ij} =$ capacidad requerida tarea i del agente j.
$c_{ij} =$ costo de asignar la tarea i al agente j.

las variables de decisión X_{ij}:

$$X_{ij} = \begin{cases} 1, & \text{si la tarea } i \text{ es asignada al agente } j. \\ 0, & \text{en otro caso.} \end{cases}$$

Se quiere:

$$\text{Minimizar } Z = \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} X_{ij}$$

Sujeto a:

$$\sum_{i=1}^{n} a_{ij} X_{ij} \leq b_j; \ \forall j = 1, ..., m \tag{2.1}$$

$$\sum_{j=1}^{m} X_{ij} = 1; \ \forall i = 1, ..., n \tag{2.2}$$

$$X_{ij} \in \{0, 1\}; \ \forall j = 1, ..., m; \ i = 1, ..., n \tag{2.3}$$

- El conjunto de restricciones 2.1 asegura que no se exceda la capacidad del agente j.
- El conjunto de restricciones 2.2 garantiza que cada tarea sea asignada a un sólo agente.
- Puesto que las variables son binarias 2.3 cada tarea es asignada a uno y sólo un agente.

Como puede observarse en el modelo anterior la función objetivo es del tipo minimizar.

Al generar una solución "X" para este problema está pueden ser:

- No factible: no cumple todas las condiciones del problema.
- Factible: cumple todas las condiciones del problema.
- Óptima: es aquella solución que cumple todas las condiciones y además es la mejor existente en todo el conjunto de combinaciones posibles.

El problema ha sido estudiado con diferentes métodos, debido a su complejidad matemática, entre los que están; ***algoritmos exactos*** que son aquellos que logran alcanzar la solución óptima, sin embargo, el tiempo computacional puede llegar a ser muy grande, ***algoritmos de aproximación*** son útiles para encontrar soluciones que está demostrado que son de calidad y cuyos tiempos de ejecución se encuentran acotados por cotas conocidas y los ***métodos heurísticos*** donde las soluciones son obtenidas de métodos imaginativos y creativos basados en la experiencia y conocimientos previos que proporcionan una solución de buena calidad en un tiempo razonable.

A continuación se presenta una revisión bibliográfica del estado del arte del *PAG*

El *PAG* fue introducido por De Maio, A. y C. Roveda en (1971)[18], sin embargo es hasta (1975) cuando Ross y Soland le dan ese nombre [61]. El *PAG* es un problema de optimización combinatoria que pertenece a la clase *NP-Difícil* Sahni y Gonzalez (1976) [63], Fisher *et al.* (1986) [22] e incluso el problema de encontrar un conjunto de asignación de tareas factible es NP-Completo(Fisher y Jaikumar, 1981)[21], ha sido de gran interés en la comunidad científica e industrial.

Ross y Soland (1977) modelan problemas de ubicación de instalaciones como un problema de asignación generalizada [62], Avraham Shtub (1989) muestran que el problema de formación de grupo de células de tecnología es equivalente al problema de asignación generalizada [68], Gottlieb *et al.* (1990), analizan tres clases de desigualdades válidas basadas en múltiples restricciones de mochila que derivan para el *PAG*, las propiedades generales de la faceta que define las desigualdades se discuten y para un caso especial, el casco convexo se caracteriza completamente, prueban que una solución fraccional básica de relajación de programación lineal puede ser eliminada por una faceta que se define como una desigualdad asociada con una restricción de mochila [25], Osman (1995) [52] presentó una perspectiva de varias aplicaciones en la vida real, Barrie M. Baker *et al.* (1999) muestran una relación aproximada entre las distancias del ruteo de vehículos y los valores del *PAG*, combinando la relajación lagrangiana en un método relativamente sencillo

para obtener soluciones del *PAG* cercanas al óptimo. [7].

El *PAG* ha sido estudiado ampliamente se han utilizado diferentes algoritmos para estudiar su comportamiento, en las siguientes tablas se mencionan algunos en orden cronológico.

Técnica	Año	Nombre del Artículo	Autores	comentarios
R & A	1975	A branch and bound algorithm for the generalized assignment problem	[61]	
Relajación PL y R & A	1977	Modeling facility location problems as generalized assignment problems	[62]	
Modificación de ajuste de subgradiente y R & A	1979	An effective subgradient algorithm for generalized assignment problem	[34]	Dos estados de un algoritmo heurístico usa una modificación de una aproximación de subgradiente y desarrollan un R & A para su solución
RL	1981	Bound and bound algorithm for the zero-one multiple knapsack problem	[43]	Se basa en el Algoritmo (R & A) a través de un esquema de acotamiento con (RL)(RS) obtienen mejores resultados.
R & A	1986	A multiplier adjustment method for the generalized assignment problem	[22]	Implementación de (R & A) en el cual las cotas son obtenidas por una relajación lagrangiana con multiplicadores de conjunto
RL	1986	A new Lagrangian relaxation approach to the generalized assignment problem	[31]	
RL	1989	An Improved Dual Based Algorithm for the Generalized Assignment Problem	[26]	Algoritmo basado en un procedimiento de ascenso dual lagrangiano mejorado para resolver el dual lagrangiano, una estrategia compleja de acotamiento que reduce la búsqueda en el árbol y adiciona una restricción sustituta para el problema relajado.
Una heurística para el NLPAG (Relajación y R & A)	1989	Generalized Assignment with Nonlinear Capacity Interaction	[44]	
Heurística RL	1992	A linear relaxation heuristic for the generalized assignment problem	[72]	
Algoritmo de aproximación (makespan)	1993	An approximation algorithm for the generalized assignment problem	[67]	Generan un algoritmo de aproximación con base al tiempo de realización de las tareas para calendarizarlas y obtener soluciones factibles.
Aproximacion por agregacion y desagregacion de columnas	1993	Solving large scale generalized assignment problems An aggregation/disaggregation approach	[28]	
HVBP	1994	A Rigorous Computational Comparison of Alternative Solution Methods for the Generalized Assignment Problem	[3]	Compara el algoritmo heurístico de variable de búsqueda en profundidad con el método de Martello y Tod
Heurística de partición de conjuntos	1994	A set partitioning heuristic for the generalized assignment problem	[12]	
Método de multiplicador de ajuste (MMSH)	1994	An steepest multiplier adjustment method for the generalized assignment problem	[33]	
Variable de busqueda profunda (VBP)	1994	A robust heuristic for the Generalized Assignment Problem	[55]	
Inicio con arepentimiento y refinamiento codicioso (HH, HPAG y VBP)	1995	A hybrid heuristic for the generalized assignment problem	[4]	
Hibrido RS y BT	1995	Heuristics for the generalised assignment problem: simulated annealing and tabu search approaches	[52]	
BT	1995	Tabu search for the multilevel generalized assignment problem	[36]	
RL	1996	Relaxation heuristics for a generalized assignment problem	[41]	

Cuadro 2.1: Estado del arte

Técnica	Año	Nombre del Artículo	Autores	comentarios
AG	1997	A genetic algorithm for the generalised assignment problem	[13]	
R & P	1997	A branch and price algorithm for the generalized gssignment problem	[65]	
R & P	1997	A branch-and-price algorithm for the generalized assignment problem	[65]	
RT	2001	A two-phase path relinking algorithm for the generalized assignment problem	[2]	Dos fases del reencadenamiento de trayectorias combina la solución del programa lineal, cortes y búsqueda local para obtener una población de soluciones con una diversidad guiada aplicando en la segunda fase el PR a los individuos de la población obtenida.
BT	2001	A tabu search heuristic for the generalized assignment problem	[19]	Muestra un BT con medida de ajuste dinámico del peso de la penalización en caso de solución no factible.
BT	2001	A dynamic tabu search for large-scale generalised assignment problems	[29]	
R & C	2001	A family of inequalities for the generalized assignment polytope	[17]	
Modelo estocasico	2001	Generating experimental data for the generalized assignment problem	[60]	
(PBCAA) (MMSH) (BT)	2002	Heurísticas adaptativas para el problema de asignación generalizada	[58]	
AG	2002	A constructive genetic algorithm for the generalised assignment problem	[40]	
CE -> (CEPM,CEPC)	2003	Independent and cooperative parallel search methods for the generalized assignment problem	[5]	utiliza algoritmo de cadena de expulsión de Yagiura generan las soluciones de un multi-comienzo CE y un comparativo paralelo CE
RL y R & A	2003	Solving the generalized assignment problem: an optimizing and heuristic approach	[49]	
Hibridación GA	2004	An improved hybrid genetic algorithm for the generalized assignment problem	[20]	Muestra dos alternativas de inicialización heurísticas, una para modificación del esquema de selección y remplazo para manejo de soluciones infactibles y la otra para el operador de mutación.
R & A método de subgradiente (MS) relajación lagrangiana (RL)	2004	Effective algorithm and heuristic for the generalized assignment problem	[27]	se presenta un nuevo R & A en el cual utilizan un método de subgradiente para resolver el dual de la relajación lagrangiana.
PBCAA MMSH BT	2004	Métodos de solución de problemas de asignación de recursos sanitarios	[42]	
BT y CE	2004	An ejection chain approach for the generalized assignment problem	[74]	
AG y RS	2005	Algoritmo memético aplicado a la solución del problema de asignación generalizada	[51]	
R & C & P	2005	Stabilized branch-and-cut-and-price for the generalized assignment problem	[53]	
R & C	2006	Exact solutions to a class of stochastic generalized assignment problems	[1]	Ocupan su propio R & C para resolver el EPAG o denominado PAG estocástico primero rama y luego corte PRDC, PCDR primero corte después rama, RCS rama y corte simultaneo
Algoritmo de aproximación	2006	A (1–1/e) - approximation algorithm for the generalized assignment problem	[50]	
RL	2006	Solving the generalized assignment problem by column enumeration based on lagrangian reduced costs	[11]	
Algoritmo de aproximación "técnica de relación local"	2006	An efficient approximation for the generalized assignment problem	[15]	
RTCE	2006	A path relinking approach with ejection chains for the generalized assignment problem	[75]	
ACA	2007	Artificial bee colony algorithm and its application to generalized assignment problem	[9]	Adaptacion del ACA al GAP
AG y RS	2007	Solving the Assignment problem using Genetic Algorithm and Simulated Annealing	[64]	AG con cruza PMX utilizando RS con criterio de metrópoli exponencial
Heuristica relajación (PL) con un orden especial de conjuntos.	2007	An LP-based heuristic procedure for the generalized assignment problem with special ordered sets	[23]	
RL y una regrecion voras	2007	Lagrangian relaxation guided problem space search heuristics for generalized assignment problems	[30]	
BIPAG	2007	An efficient solution to biobjective generalized assignment problem	[78]	
R & A	2009	A column generation heuristic for a dynamic generalized assignment problem	[46]	
Una heurística asintóticamente óptima para un programa lineal	2009	The generalized assignment problem with flexible jobs	[56]	

Cuadro 2.2: Estado del arte

Técnica	Año	Nombre del Artículo	Autores	comentarios
Aproximacion basada en principios de dominancia	2009	A new heuristic approach for the large-scale Generalized assignment problem	[8]	
Escala muy larga de vecindario (EMLV)	2009	Local search intensified: very large-scale variable neighborhood search for the multi-resource generalized assignment problem	[45]	
ED VNS	2009	Differential evolution algorithms for the generalized assignment problem	[71]	
BTRA (BT y R & A)	2010	A hybrid tabu search/branch & bound approach to solving the generalized assignment problem	[73]	
R & P Dantzing-wolf	2010	A computational study of exact knapsack separation for the generalized assignment problem	[6]	
RL y una heurística voras	2010	Lagrangian heuristic for a class of the generalized assignment problems	[37]	
R & P	2010	A computational study of exact knapsack separation for the generalized assignment problem	[6]	
BT y R & A	2010	Solving the generalized assignment problem: a hybrid Tabu search/branch and bound algorithm	[73]	
AG	2012	The equilibrium generalized assignment problem and genetic algorithm	[38]	Heurístca del AG para el EGAP.
Sistema de colonia de hormigas difuso (SCHD)	2012	Optimización de un problema de asignación generalizada parte de un algoritmo de colonia de hormigas que incorpora un mecanismo de difusión	[69]	
RL y R & A con reglas de reparacion	2012	A exact method with variable fixing for solving the generalized assignmet problem	[54]	
Dos algoritmos exactos para el PAG	2013	Two exact algorithms for the generalized assignment problem	[57]	presenta un algoritmo O (n4) en tiempo y O (n) espacio para calcular una asignación generalizada entre dos conjuntos, se modifica el método húngaro para obtener un nuevo algoritmo de asignación generalizada el cual es aplicado a una gráfica bipartita previamente construida y también improvisan un algoritmo O(n3)para el problema de la asignación de la capacidad limite.
(PAG -MQ)	2013	The generalized assignment problem with minimum quantities	[35]	
ACA	2013	Solving fuzzy multiple objective generalized assignment problems directly via bees algorithm and fuzzy ranking	[70]	
YAGER'S Ranking Method (BFS)	2014	Solution of generalized fuzzy assignment problem with restriction on costs under fuzzy environment	[32]	
R & A	2015	A transportation branch and bound algorithm for solving the generalized assignment problem	[48]	Realizan una relajación del problema para obtener la cota mas baja, Selecciona la fila más restringida para obtener variables de bifurcación, ramifica seleccionando variables hasta regresar la mejor solución de transporte factible.
AG	2015	A scalable parallel genetic algorithm for the Generalized Assignment Problem	[39]	
ED con tres algoritmos de búsqueda para cambio de vecindario	2016	Improved differential evolution algorithms for solving generalized assignment problem	[66]	

Cuadro 2.3: Estado del arte

Capítulo 3

Métodos Base

Los algoritmos pertenecientes a la rama de inteligencia de partículas ***IP***, los cuales imitan el comportamiento colectivo descentralizado en un sistema auto organizado, basado en una población de agentes *(individuos)* los cuales interactúan entre ellos y con su ambiente.

En la presente sección se describen los algoritmos de luciérnagas ***AL***, búsqueda gravitacional ***ABG*** y el método de composición musical ***MCM***, ya que éstos serán empleados como técnicas base para el desarrollo de estrategias de solución para el *PAG* en este trabajo.

Se enfatiza en que estos procedimientos son del tipo poblacional, lo cual involucra partir de una población inicial de individuos. Cada individuo representa una posible solución del problema; esta se puede generar de manera aleatoria o bien con base en alguna estrategia que explote las características del problema.

De manera general se obtiene una población inicial producida de manera aleatoria, sin embargo, esta estrategia tiene alta probabilidad de generar soluciones infactibles.

En este trabajo se propone partir de generar una población inicial que permita a los algoritmos iniciar explorando regiones prometedoras, para lo cual se aprovecharán las propiedades de la relajación lineal y la idea de los métodos glotones lo que permite dirigir la búsqueda a regiones prometedoras desde un inicio.

Si bien, la población inicial de soluciones podría tener algunas no factibles, se busca satisfacer la mayor cantidad de restricciones, lo que contribuirá a que el algoritmo requiera un menor esfuerzo para alcanzar la región factible.

A continuación se describen los métodos de partículas empleados en esta investigación, en el siguiente capítulo se describen con detalle las adaptaciones realizadas a los mismos para resolver el *PAG*.

3.1. Algoritmos empleados

3.1.1. Algoritmo de luciérnagas

El *AL* fue desarrollado por Xin-Shen Yang de la Universidad de Cambridge en (2007) [77].

Pertenece a la clase *IP*, este algoritmo imita el comportamiento social (intermitente) de luciérnagas o bichos de luz, que emiten destellos cortos y rítmicos con dos principios fundamentales:

- Comunicación (atraer compañeros de apareamiento).
- Atracción de la presa potencial.

Por lo que algunas luciérnagas tropicales pueden sincronizar sus destellos, formando así un comportamiento biológico auto organizado.

Existen dos cuestiones importantes del *AL* asociado con la función objetivo, la variación de la intensidad luminosa y el brillo [77]:

- La variación de la intensidad de la luz I(r) varía de acuerdo a la ley del inverso cuadrado.
$$I(r) = \frac{I_s}{r^2} \tag{3.1}$$

- Donde I_s es la intensidad de la fuente, para una media dada con un ajuste del coeficiente de absorción de luz γ, la intensidad de la luz I varía con la distancia r. Esto es
$$I = I_0 e^{\gamma r} \tag{3.2}$$

- Donde I_0 es la intensidad de luz original, en el caso singular donde $r = 0$ en la expresión $I(r) = \frac{I_s}{r^2}$, el efecto combinado de la ley del cuadrado inverso y la absorción puede aproximarse de la forma gaussiana mostrada acontinuación.
$$I = I_0 e^{\gamma r^2} \tag{3.3}$$

- La formulación de atractivo β

$$\beta = \beta_0 e^{\gamma r^2} \tag{3.4}$$

- La distancia entre dos luciérnagas cualesquiera i y j ubicadas en los puntos x_i y x_j respectivamente, es la distancia cartesiana entre la dimensión $k = (k_1, ..., k_n)$.

$$d_{i,j} = ||x_i - x_j|| = \sqrt{\sum_{k=1}^{n} (x_{i,k} - x_{j,k})^2} \tag{3.5}$$

En [77] idealizan las características de los destellos de las luciérnagas para desarrollar el AL.

El movimiento de una luciérnaga i con localización x_i atraída por otra luciérnaga j más brillante y localización x_j esta determinada por:

$$x_i(t+1) = x_i(t) + \beta_0 e^{\gamma d_{i,j}^2}(x_j - x_i) + \alpha\varepsilon_i \tag{3.6}$$

Donde t indica el instante o iteración correspondiente, α es un parámetro aleatorio, ε_i es un vector de números aleatorios uniformemente distribuidos, de forma simple cada elemento de ε_i puede ser reemplazado por un número aleatorio uniformemente distribuido entre [0,1].

3.1.2. Pseudocódigo *AL*

Algoritmo 1 Procedimiento general Luciérnagas [77]

Función objetivo $f(x)$, $x = (x_1, ..., x_d)^T$
Generar una población inicial de luciérnagas $x_i = (1, ..., n)$
Calcular la intensidad luminosa I_i **y** x_i está determinada por $f(x)$
Definir el coeficiente de absorción de luz γ
while t < Generación máxima **do**
  for i= 1 : n **do**
    for j= 1 : n **do**
      if $I_i < I_j$, mover luciérnaga i hasta j **then**
        Variar el atractivo con la distancia r a través $exp[-r]$
        Evaluar la nueva solución y actualizar la intensidad luminosa
      end if
    end for
  end for
  Clasificar las luciérnagas y encontrar el mejor global actual $l*$
end while
resultados de post-procesado y visualización

Nota: para este este algoritmo en particular la distancia r es calculada como la distancia euclidiana entre dos luciérnagas

3.2. Método de gravitación universal

La ley de Gravitación Universal fue publicada en 1687 por Isaac Newton. El *ABG* fue introducido por Esmat Rashedi *et al.* (2009) [59] donde se muestra un algoritmo de búsqueda poblacional basado en las reglas de la gravitación universal donde se da una breve descripción de la fuerza gravitacional para proporcionar un medio apropiado y seguido por la explicación de *ABG*.

3.2.1. Ley de la gravitación

La gravitación es la tendencia de las masas a acelerar una hacia otra, ésta es una de las cuatro interacciones fundamentales en la naturaleza (las otras son: la fuerza electromagnética, la fuerza nuclear débil y la fuerza nuclear fuerte). Por lo que, cada partícula en el universo atrae a cualquier otra partícula con una "fuerza gravitacional" [59].

La forma de la fuerza gravitacional de Newton es llamada "acción distancia", significa cambios de gravedad entre partículas separadas sin cualquier intermediario y sin demora alguna.

La fuerza gravitacional entre dos partículas es directamente proporcional al producto de sus masas M_1 y M_2 e inversamente proporcional al cuadrado de las distancias R entre ellas.

$$F = \frac{GM_1M_2}{R^2} \tag{3.7}$$

Donde F es la magnitud de la fuerza gravitacional, G es la constante gravitacional, M_1 y M_2 son las masas de la primera y segunda partícula respectivamente y R es la distancia entre éstas.

La segunda ley de Newton dice que cuando una fuerza, "F", es aplicada a una partícula, su aceleración "a", depende sólo de la fuerza y su masa "M".

$$a = \frac{F}{M} \tag{3.8}$$

Hay una fuerza de atracción entre todas las partículas del universo, donde el efecto de la partícula más grande y la partícula más pequeña es mayor, por lo que, un incremento en la distancia entre dos partículas significa decrecimiento en la fuerza de gravedad entre ellas.

Debido al efecto de disminuir la gravedad, el valor actual de la constante gravitacional depende del estado actual del universo, la siguiente fórmula describe el decrecimiento de la contante gravitacional, "G", con el estado:

$$G(t) = G(t_0)x(\frac{t_0}{t})^{\beta}, \quad \beta < 1 \tag{3.9}$$

Donde $G(t)$ es el valor de la constante gravitacional en el tiempo t, $G(t_0)$ es el valor de la constante gravitacional en el intervalo de tiempo t_0.

En el ABG cada masa(Agente) tiene cuatro especificaciones:

> a) Posición, b) Masa de inercia, c) Masa gravitacional activa, d) Masa gravitacional pasiva.

La posición de la masa corresponde a una solución del problema y las masas de inercia y gravitacional son determinadas utilizando la función objetivo o función aptitud, en otras palabras cada masa representa una solución y el algoritmo navega ajustando correctamente las masas gravitacionales y la inercia, con el transcurso de tiempo, esperando que las masas sean atraídas por las masas más pesadas, éstas representan una mejor solución en el espacio de búsqueda.

3.2.2. Pseudocódigo ABG

Algoritmo 2 Procedimiento ABG básico

1: Generar una población inicial aleatorizada
2: **while** no se cumpla criterio de paro **do**
3: Evaluar la función aptitud para la población
4: Calcular masa, fuerza y aceleración para la población, Ec.(9, 10 y 11)
5: Actualizar la velocidad y posición de la población, Ec. (12 y 13)
6: **end while**
7: Si el criterio de terminación es alcanzado se muestra la mejor solución obtenida

3.3. Método de composición musical

El *MCM* es un algoritmo cultural inspirado en una analogía entre el proceso de optimización y el proceso creativo de composición musical en un entorno sociocultural fue planteado Mora Gutiérrez et. al (2012) para resolver un problema de optimización con restricciones [47]. Para el diseño y el desarrollo del algoritmo *MCM* se utilizaron las similitudes entre los procesos de composición musical y de optimización.

La idea principal del algoritmo se basa en el proceso creativo que combina la imaginación, elementos musicales y conocimientos a fin de obtener una obra musical que cumpla con cierto estándar estético. Por lo que, el proceso de composición es más complejo que la simple asociación de elementos, ya que involucra los sistemas creativos sociocultural y personal para la generación de un producto artístico[47].

La estructura básica del algoritmo es la siguiente: inicialmente, se genera una sociedad artificial de Nc compositores y se definen las reglas de interacción entre los agentes que la conforman. Entonces, para cada uno de los compositores en la sociedad, aleatoriamente, se crea un conjunto Ns de temas que se registran en la partitura asociada a ese compositor $P = (P_1, ..., P_i)$ la partitura servirá como la memoria del compositor, posteriormente y hasta satisfacer el criterio de paro, se realiza lo siguiente:

a) Se actualizan los vínculos entre los compositores de la sociedad.

b) Cada uno analiza la información recibida de los demás compositores y selecciona datos que toma del entorno a esto se le denomina ideas adquiridas socialmente $ISC = (ISC_1, ..., ISC_i)$.

c) El compositor i construye su conocimiento $Km = (Km_1, ..., Km_i)$ uniendo su conocimiento con la información obtenida $KM = P \bigcup ISC$.

d) Cada compositor genera una nueva melodía $(x_i, nuevo)$ a partir de su conocimiento y sus destellos de genialidad, determinando el grado de satisfacción alcanzado por $x_i, nuevo$

e) Finalmente con base en el grado de satisfacción, debe decidir si $x_i, nuevo$ remplazará a algún elemento de su partitura.

3.3.1. Pseudocódigo MCM

Algoritmo 3 Algoritmo MCM básico [47]

```
1:  Crear una sociedad artificial con reglas de interacción entre los agentes.
2:  for cada uno de los agentes de la sociedad do
3:      Generar aleatoriamente una obra musical (Para esta actividad se con-
    sidera la información sobre la instancia a resolver)
4:  end for
5:  while No se satisface el criterio de paro do
6:      Actualizar los vínculos de la red social.
7:      Intercambiar información entre agentes.
8:      for cada uno de los agentes de la sociedad do
9:          Actualizar la matriz de conocimiento.
10:         Generar y evaluar una nueva melodía (x_{⋆,nevo})
11:         Actualizar la partitura .
12:     end for
13:     Construir el conjunto de soluciones encontradas con la mejor melodía
    de cada compositor.
14: end while
```

Capítulo 4

Implementación

En el presente capítulo, se presentan las modificaciones y adaptaciones hechas a los algoritmos AL, ABG y MCM para la resolución del PAG, en la primera parte se describen la forma de codificación y generación de soluciones iniciales, las cuales son comunes a los métodos propuestos, posteriormente se describen y se detallan las propuestas desarrolladas.

4.1. Codificación

Para realizar la adaptación de un algoritmo y que éste trabaje de manera adecuada, un elemento fundamental es la codificación, ya que de ésta dependerá que el algoritmo tenga un buen desempeño. Como se mencionó en el capítulo dos, los algoritmos que se desarrollan en este trabajo son del tipo poblacional y dependen de una población inicial de soluciones, para que en ésta los individuos iteractúen entre ellos y su ambiente, los cuales mejorarán con el paso del tiempo.

Como se mostró en el capítulo uno el ***PAG*** consta de dos conjuntos, una de tareas de cardinalidad "n" y otro de agentes de cardinalidad "m". Con base en ello se representa a un individuo como un vector de tamaño $(n + m + 4)$. Dicho vector se encuentra compuesto por dos partes: en la primera parte las primeras n celdas representan el número de tareas a realizar y en su interior el número del agente asignado para su realización, suponiendo que:

- $n = 15$ tareas.
- $m = 5$ agentes.

T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	3	2	3	5	1	4	4	5	1	4	2	3	3	2	3

Cuadro 4.1: Individuo parte 1

Esta parte del vector muestra la asignación de tareas a agentes, por lo que en la segunda parte las $(m+4)$ posiciones del individuo, se componen de la manera siguiente:

F.O.	#inconsistencias	A1	A2	A3	A4	A5	Cap. Sup.	Cap. Disp.
267	1	-19	-2	30	-9	-4	30	34

Cuadro 4.2: Individuo parte 2

El objetivo del *PAG* es minimizar la suma de todos los costos de asignación, la cual observamos en las primeras n posiciones del individuo, el costo de esa asignación se muestra en la casilla $n+1$, en caso de que alguno de los individuos supere su capacidad será una inecuación en la asignación, por lo que en la casilla $n+2$ se muestra el número de inecuaciones, las "m" casillas siguientes representan a cada uno de los agentes, si el valor es negativo indica la capacidad disponible, mientras que si el valor es positivo significa que su capacidad fue superada en esa medida, la penúltima casilla muestra el total de la capacidad superada y la última muestra el total de la capacidad disponible; de tal manera que un individuo de la población sera visto como se muestra en el cuadro 4.3.

3	2	3	5	1	4	4	5	1	4	2	3	3	2	3	267	1	-19	-2	30	-9	-4	30	34

Cuadro 4.3: Individuo

De manera general, las técnicas en la literatura parten de una población inicial obtenida de manera aleatoria, pero debido a que las asignaciones producen soluciones de mala calidad ya que puede incurrir en violaciones del sistema de restricciones.

Cabe hacer notar que, las técnicas desarrolladas en este trabajo aprovechan las propiedades del problema para redirigir la búsqueda a regiones prometedoras desde un inicio. Por lo cual, la población inicial se obtenida por una combinación de tres técnicas para generar soluciones: la primera será obtenida por un algoritmo glotón, la segunda por la relajación de un programa lineal y la última de manera aleatoria. Lo cual, le permite a las técnicas desarrolladas partir de una población inicial donde la mayoría de sus individuos cumplen con la mayor cantidad de resticciones pero con soluciones diversas.

Esta propuesta aumentará los siguientes dos parámetros a los algoritmos AL, ABG y MCM:

- P_g = es el porcentaje de los individuos de la población inicial a generar por un algoritmo glotón.

- P_a = es el porcentaje de los individuos de la población inicial producidos de manera aleatoria.

Con el objetivo de ilustrar lo anterior considérese el ejemplo siguiente: Dada una población de 10 individuos y dado $P_g = 0.5$ y $P_a = 0.5$ significa que 5 individuos serán obtenidos por un algoritmo glotón, tres individuos serán obtenidos de manera aleatoria y los últimos 2 individuos serán obtenidos por una relajación de un programa lineal.

Los algoritmos que serán utilizados para generar la población inicial se muestran a continuación.

4.2. Método de generación de soluciones iniciales

4.2.1. Algoritmo Glotón

Los algoritmos voraces se utilizan generalmente para resolver problemas de optimización.

Este algoritmo se construye en etapas, cada una de éstas se va agregando un elemento el cual se considera como el mejor en ese momento y es agregado a la solución parcial, las decisiones tomadas no se revisan debido a que se selecciona el mejor individuo sin pensar en futuras consecuencias y no garantizan la solución óptima.

Pseudocodigo algoritmo glotón

Algoritmo 4 Procedimiento algoritmo glotón

Se genera una matriz del producto punto de la matriz de costo por la matriz capacidad = a
while Tareas asignadas $\leqslant$ n ó contador $< 2 * n$ **do**
    Se genera una matriz de probabilidad de selección
    Se van seleccionando las tareas y su respectivo agentes de manera aleatoria con la matriz de probabilidad de selección guardándose la asignación en el vector solución
end while
Se revisa el vector solución
if Si el vector solución no asignó todas las tareas **then**
    las tareas que no fueron asignadas se les asigna un agente de manera aleatoria
end if

4.2.2. Relajación PL

El problema de programación lineal que se obtiene al omitir todas las restricciones enteras ó variables 0-1 se llama relajación de programación lineal para la programación entera.

La Programación Lineal asume que las variables de decisión son continuas. Sin embargo, en muchas aplicaciones, los valores fraccionarios pueden no tener sentido, los problemas de programación lineal con enteros son más difíciles de resolver que los de programación lineal continua. ¿ Por qué no resolver todos los problemas como problemas de programación lineal y redondear las respuestas a los enteros más cercanos?, desafortunadamente, esto genera dos problemas:

- La solución redondeada puede no ser factible.
- El redondeo puede no dar una solución óptima.

Pseudocodigo Relajación PL

Algoritmo 5 Procedimiento Relajación PL

Se relaja el problema en un PL
Se resuelve el PL por el método de larga escala, el cual fija algunas variables
Las variables faltantes se fijarán de manera aleatoria

4.3. Implementación del AL

Una de las principales adaptaciones realizadas al algoritmo es la introducción de los dos parámetros P_g y P_a los cuales servirán para diseñar la población inicial.

Para determinar la intensidad luminosa de una luciérnaga I_0, a diferencia del algoritmo original donde ésta se determina por el paisaje de la función objetivo, en este caso se consideran también el número de agentes que superan su capacidad, el total de la capacidad superada y el total de la capacidad disponible, a través de las reglas de manejo de restricciones de Coello(2002)[14], el siguiente pseudocódigo muestra cómo se determinan los pesos.

Pseudocódigo determinación pesos para reglas de manejo de restricciones

Algoritmo 6 Determinación pesos para reglas de manejo de restricciones

Entrada población(i)
Se generan cuatro vectores v_1,v_2,v_3,v_4
En v_1 se guarda los valores de Fo de cada individuo
En v_2 se guarda el número de restricciones que no satisface el individuo
En v_3 se guarda la capacidad total superada por el individuo por no satisfacer las restricciones
En v_4 la capacidad total disponible de las restricciones satisfechas
se calculan los valores máximos y mínimos de cada uno de los vectores
Se calculan cuatro variables xa, xb, xc, xd
if $minv_1 - maxv_1 == 0$ **then**
    $xa = 1$
else
    $xa = minv_1 - maxv_1$
end if
if $minv_2 - maxv_2 == 0$ **then**
    $xb = 1$
else
    $xb = minv_2 - maxv_2$
end if
if $minv_3 - maxv_3 == 0$ **then**
    $xc = 1$
else
    $xc = minv_3 - maxv_3$
end if
if $minv_4 - maxv_4 == 0$ **then**
    $xd = 1$
else
    $xd = minv_4 - maxv_4$
end if
Se generan los pesos:
$VpFO = (v_1 - max(v_1))/(xa)$
$VpInf = (v_2 - max(v_2))/(xb)$
$VpCapsup = (v_3 - max(v_3))/(xc)$
$VpCapdis = (v_4 - max(v_4))/(xd)$

Pseudocódigo calculo Intensidad luminosa

Algoritmo 7 Calculo de Intensidad Luminosa por manejo de restricciones

```
Datos de entrada VpFO,VpInf,VpCapsup y VpCapdis
for i=1:población do
    for j=1:población do
        if VpInf_i >= VpInf_j then
            if VpInf_j > VpInf_j then
                I_1(i,j) = (1/poblacion^2)
            else
                if Es factible pero VpInf_j == 0 then
                    if VpFO_i >= VpFO_j then
                        if VpFO_i == VpFO_j then
                            if VpCapdis_i > VpCapdis_j then
                                I_1(i,j) = (1/poblacion^3)
                            else
                                I_1(i,j) = 0
                            end if
                        else
                            I_1(i,j) = (1/poblacion)
                        end if
                    else
                        I_1(i,j) = 0
                    end if
                end if
                if VpCapsup_i > VpCapsup_i then
                    I_1(i,j) = ÷1/poblacion^3
                else
                    I_1(i,j) = 0
                end if
            end if
        else
            I_1(i,j) = 0
        end if
    end for
end for
Por lo que la Intensidad para cada individuo será calculado de la matriz
de intensidad
for i=1:población do
    I(i,1) = sum(I_1(i,:))/poblacion
end for
```

En el algoritmo original la distancia entre dos luciérnagas cualesquiera ubicadas respectivamente x_i y x_j, es la distancia cartesiana véase la ecuación 3.5. Para ésta implementación del algoritmo se tomará en cuenta también la distancia de Hamming quedando así la siguiente formulación.

$$d_{ij} = \sqrt{(Hamming(x_i \neq x_j) * ||x_i - x_j||) + 1} \tag{4.1}$$

Para este algoritmo se desarrollaron seis implementaciones diferentes las cuales se muestran a continuación.

4.3.1. AL1

AL1: consta de los siguiente pasos.

1. Se produce una población inicial con base a los factores P_g y P_a.
2. Se calcula la $I(t)$ y el brillo $\beta(t)$ de la luciérnaga siguiendo las reglas de manejo de restricciones.
3. Determinar cuántas posiciones serán cambiadas usando la ecuación de movimiento.
4. El cambio de las posiciones se realiza de manera aleatoria.
5. Se evalúa la nueva solución y si cumple el criterio de aceptación se acepta, si no, se rechaza.

Pseudocódigo AL1

Algoritmo 8 AL 1

Características del problema a resolver:
Función objetivo $f(x)$, $x = (x_1, ..., x_d)$
Definir P_g,P_a
Generar una población inicial de luciérnagas $x_i = (1, ..., n)$ determinado por los algoritmos
Calcular la intensidad luminosa I_i **y** x_i esta determinada por $f(x_i)$
Definir γ, α
while t < Generación máxima **do**
  for i= 1 : n **do**
    for j= 1 : n **do**
      if $I_i < I_j$, mover luciérnaga i hasta j **then**
        Se calcula la f(dominancia)
        Variar el atractivo con la distancia r a través $exp[-\gamma r]$
        Se determina la cantidad de posiciones para cambio de etiqueta y se realizan cambios con base a matriz de probabilidad
        Evaluar la nueva solución y actualizar la intensidad luminosa
      end if
    end for
  end for
  Clasificar las luciérnagas y encontrar el mejor global actual $f(x)*$
end while
resultados de post procesado y visualización

4.3.2. AL2

1. Se toma la mejor solución de la población.
2. Se hace una búsqueda local y se realiza un cambio de vecindario.
3. El cambio del vecindario se realiza intercambiando dos etiquetas diferentes.
4. Se evalúa la nueva solución y si cumple el criterio de aceptación se acepta, si no se rechaza.

Pseudocódigo AL2

Algoritmo 9 AL 2

Características del problema a resolver:
Función objetivo $f(x)$, $x = (x_1, ..., x_d)$
Definir P_g,P_a
Generar una población inicial de luciérnagas $x_i = (1, ..., n)$ determinado por los algoritmos
Calcular la intensidad luminosa I_i **y** x_i esta determinada por $f(x_i)$
Definir γ, α
while t < Generación máxima **do**
    Se toma la mejor luciérnaga
    Se genera una nueva población con base a la mejor luciérnaga de la población anterior, con cambio de vecindario de dos etiquetas
    for i= 1 : n **do**
        for j= 1 : n **do**
            if $I_i < I_j$, mover luciérnaga i hasta j **then**
                Se calcula la f(dominancia)
                Variar el atractivo con la distancia r a través $exp[-\gamma r]$
                Evaluar la nueva solución y actualizar la intensidad luminosa
            end if
        end for
    end for
    Clasificar las luciérnagas y encontrar el mejor global actual $g*$
end while
resultados de post procesado y visualización

4.3.3. AL3

1. Si el atractivo de la luciérnaga con la que es comparada es mejor se realiza el movimiento.

2. Se realizan los puntos 2 y 3 de la técnica 1
3. Se realiza un cambio de vecindario.
4. Se evalúa la nueva solución y si cumple el criterio de aceptación se acepta, si no, se le asigna una probabilidad de aceptación.

Pseudocódigo AL3

Algoritmo 10 AL 3

Características del problema a resolver:
Función objetivo $f(x)$, $x = (x_1, ..., x_d)$
Definir P_g,P_a
Generar una población inicial de luciérnagas $x_i = (1, ..., n)$ determinado por los algoritmos
Calcular la intensidad luminosa I_i **y** x_i esta determinada por $f(x_i)$
Definir γ, α
while t < Generación máxima **do**
  for i= 1 : n **do**
    for j= 1 : n **do**
      if $I_i < I_j$, mover luciérnaga i hasta j **then**
        Se calcula la f(dominancia)
        Variar el atractivo con la distancia r a través $exp[-\gamma r]$
        Se determina la cantidad de posiciones para cambio de etiqueta y se realizan cambios con base a matriz de probabilidad
        Se realiza un cambio de vecindario intercambiando dos etiquetas
        Evaluar la nueva solución y actualizar la intensidad luminosa
      end if
    end for
  end for
  Clasificar las luciérnagas y encontrar el mejor global actual $g*$
end while
resultados de post procesado y visualización

4.4. AL4

1. Se determina un factor V_1 para cada determinado número de iteraciones realizar una reducción del problema.
2. Se realizan los puntos 1 al 3 de la técnica 1.

3. Si la iteración es elegida para la reducción del problema, genera una matriz de probabilidad, se determinan qué variables quedan fijas y se resuelve el programa lineal reducido.

4. Se evalúa la nueva solución y si cumple el criterio de aceptación se acepta, si no, se le da una probabilidad de aceptación.

Pseudocódigo AL4

Algoritmo 11 AL 4

Características del problema a resolver:
Función objetivo $f(x)$, $x = (x_1, ..., x_d)$
Definir P_g, P_a, V1
Generar una población inicial de luciérnagas $x_i = (1, ..., n)$ determinado por los algoritmos
Calcular la intensidad luminosa I_i **y** x_i esta determinada por $f(x_i)$
Definir γ, α
while t < Generación máxima **do**
  Se redefine el factor V_1 para determinar la iteración en la que se hace la reducción de del problema
  for i= 1 : n **do**
    for j= 1 : n **do**
      if $I_i < I_j$, mover luciérnaga i hasta j **then**
        Se calcula la f(dominancia)
        Variar el atractivo con la distancia r a través $exp[-\gamma r]$
        Se determina la cantidad de posiciones para cambio de etiqueta y se realizan cambios con base en la matriz de probabilidad
        Evaluar la nueva solución y actualizar la intensidad luminosa
      end if
    end for
  end for
  if Si la Generación fue elegida para reducción del problema **then**
    se reduce con base a un esquema, se relaja y resuelve el programa lineal el cual fija algunas variables
    En caso de que alguna variable no haya sido a asignada la asigna con base a una matriz de probabilidad
    De la solución obtenida generar una nueva población realizando una búsqueda local
  end if
  Clasificar las luciérnagas y encontrar el mejor global actual $g*$
end while
resultados de post procesado y visualización

4.4.1. AL5

1. Se realizan los puntos 1 al 3 de la técnica 2

2. Se determina un factor V_1 para que cierto número de iteraciones realizar una reducción del problema.

3. Se genera una matriz de probabilidad, se determina que variables quedan fijas y se resuelve el programa lineal reducido de las variables que quedaron sin fijar.

4. Se evalúa la nueva solución y si cumple el criterio de aceptación se acepta, si no se le da una probabilidad de aceptación..

4.4.2. AL6

1. Se realizan los puntos 1 al 3 de la técnica 3

2. Se determina un factor V_1 para que en cierto número de iteraciones se reduzca el problema.

3. Se genera una matriz de probabilidad, se determinan que variables quedan fijas y se resuelve el programa lineal reducido de las variables que quedaron sin fijar.

4. Se evalúa la nueva solución y si cumple el criterio de aceptación se acepta, si no se le da una probabilidad de aceptación.

4.5. Implementación del ABG

Se introducen dos parámetros P_g y P_a que como se mostró en las secciones 4.1 y 4.2 para obtener la población inicial.

La constante gravitacional se actualiza en el tiempo

$$G_{(t+1)} = G_{(t)} * (\frac{t}{t_{total}})^{\beta} \tag{4.2}$$

A diferencia del algoritmo original donde la masa sería representada por el valor de la función objetivo, se genera una matriz Aptitud, usando las reglas de manejo de restricciones de Coello, se asignan pesos a elementos de la matriz para clasificar a los individuos.

4.5.1. Pseudocódigo *Determinación de masas*

Algoritmo 12 Determinación de las masas ABG

for i= 1 : población **do**
  for j= 1 : población **do**
    if i $\neq$ j **then**
      if Ambas soluciones son factibles y $Fo(i) < Fo(j)$ **then**
        $Aptitud(i,j) = 2$ y $Aptitud(j,i) = 0$
      else if Ambas soluciones son factibles y $Fo(i) = Fo(j)$ se compara la capacidad total disponible **then**
        if $captotdis(i) < captotdis(j)$ **then**
          $Aptitud_{i,j} = 1$ y $Aptitud_{j,i} = 0$
        else if $captotdis(i) = captotdis(j)$ **then**
          $Aptitud(i,j) = 0{,}5$ y $Aptitud(j,i) = 0{,}5$
        else
          $Aptitud(i,j) = 0$ y $Aptitud(j,i) = 1$
        end if
      else
        if i es factible pero j no **then**
          $Aptitud(i,j) = 2$ y $Aptitud(j,i) = 0$
        else if j es factible pero i no **then**
          $Aptitud(i,j) = 0$ y $Aptitud(j,i) = 2$
        else if Si tanto i como j son no factibles se revisan por cuánto superan la capacidad total **then**
          if $captotsup(i) < captotsup(j)$ **then**
            $Aptitud(i,j) = 1$ y $Aptitud(j,i) = 0$
          else if $captotsup(i) = captotsup(j)$ **then**
            $Aptitud(i,j) = 0{,}5$ y $Aptitud(j,i) = 0{,}5$
          else
            $Aptitud(i,j) = 0$ y $Aptitud(j,i) = 1$
          end if
        end if
      end if
    end if
  end for
end for
for i= 1 : población **do**
  Se valora la aptitud
  $valoracion(i,1) = \sum(aptitud(i,:))$
end for
Regresa el valor normalizado de las masas

Se genera una matriz de distancias, si los puntos son diferentes en los individuos comparados, se calcula la distancia euclidiana.

Para cada individuo se calcula la masa de inercia donde:

$$m_{i,j} = (G * abs(masa_i * masa_j))/(ditancia_euclidiana(i,j) + 1) \qquad (4.3)$$

Se calcula también la distancia de Hamming h_{ij} entre los individuos i y j y se calcula la fuerza de atracción donde:

$$F_a(i,:) = F_a(i,:) + (rand * m_{ij} * h_{ij}) \tag{4.4}$$

y por lo tanto.

$$aceleracion = \frac{F_a(i)}{masa(i)} \tag{4.5}$$

Se actualiza la velocidad

$$v_a = (rand * v_a) + aceleracion \tag{4.6}$$

Se identifican las posiciones que sufren alteraciones en posicionamiento, se determinan tres vecindarios para el cambio de posición.

- Cambio en un punto.
- Cambio de etiquetas entre 2 puntos.
- Se repara la solución dándole prioridad de mantener aquellos objetos que minimizan la función de costos y cuya sumatoria de capacidad requerida no rebase la capacidad disponible de los agentes, los elementos que no fueron asignados por no cumplir con las características se asignan de manera aleatoria.

Si el nuevo individuo es mejor que el anterior, se toma el nuevo individuo y se actualiza la v_a

4.6. Implementación del MCM

Se introducen dos parámetros P_g y P_a que como se mostró en las secciones 4.1 y 4.2 para obtener la población inicial.

Para genera una nueva solución en el *MCM* se realiza el siguiente procedimiento.

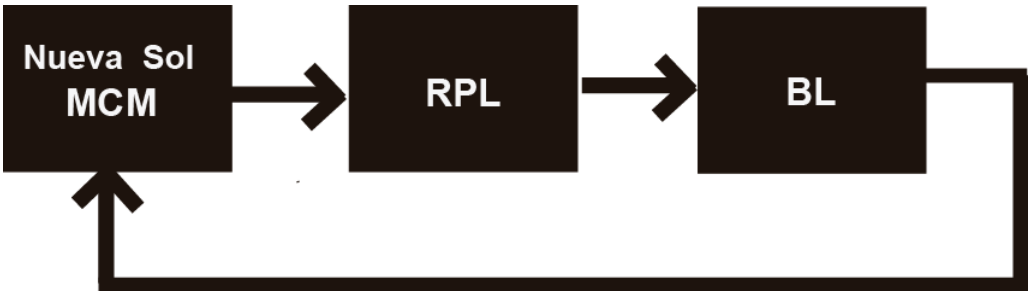


Figura 4.1: Modificación nueva solución

El *MCM* analiza las soluciones obtenidas por medio de las cuales se deduce un esquema, dando lugar la fijación de variables de un programa lineal, se resuelve el programa lineal reducido, también incluye una búsqueda local con un vecindario dinámico.

Capítulo 5

Metodología

En este capitulo se presenta la metodología experimental empleada con el objeto de caracterizar y analizar el comportamiento de las técnicas propuestas.

Por tal motivo, se toma una serie de instancias de prueba (disponible en OR-Library).

Cabe mencionar, que dichas instancias se han empleado en muchas investigaciones reportadas en la literatura para analizar y evaluar el comportamiento de métodos propuestos para resolver el PAG; ya que éstas describen una amplia gama de dificultades y particularidades para el problema mencionado.

5.1. Descripción de las instancias de prueba

Las instancias de prueba se tomaron de OR -Library [1] fueron propuestas por Chu y Beasley en [13].

Las instancias disponibles se encuentran clasificadas en cinco diferentes tipos: las cuales se describen a continuación:

Tipo A $a_{i,j}$ son enteros para U(5,25), $c_{i,j}$ es un entero para U(10,5) y $b_i = 0.6(\frac{n}{m})15 + 0.4\mathrm{R}$ donde $R = max_{i \in I} \sum_{j \in JJ_j = i} a_{i,j}$ y $I_j = min[i | C_{i,j} \leq C_{k,j}, \forall k \in I]$. [13]

Tipo B $a_{i,j}$ y $C_{i,j}$ son iguales al tipo A y b_i es un conjunto para el 70 % de los valores dados en el tipo A. [13] [76]

[1] http://people.brunel.ac.uk/ mastjjb/jeb/orlib/gapinfo.html

Tipo C $a_{i,j}$ y $c_{i,j}$ son iguales al tipo A y $b_i = 0.8 \sum_{j \in J} \frac{a_{i,j}}{m}$ [13][76]

Tipo D $a_{i,j}$ son enteros aleatorios entre [1,100], $c_{i,j} = 111 - a_{i,j} + e_1$ donde e_1 es un entero aleatrorio entre [-10,10] y $b_i = 0.8 \sum_{j \in J} \frac{a_{i,j}}{m}$. [13] [76]

Tipo E $a_{i,j} = 1 - 10 * lne_2$ donde e_2 es un número aleatorio entre (0,1], $C_{ij} = \frac{1000}{a_{ij}} - 10e_3$, donde e_3 es un número aleatorio entre [0,1], y $b_i = 0.8 \sum_{j \in J} \frac{a_{i,j}}{m}$. [76]

A continuación se describen las particularidades de los archivos existentes de instancias del *PAG* cada una es un archivo de texto plano; donde, los primeros dos valores numéricos representan a ***"m"*** (número de agentes) y ***"n"*** (número de tareas) respectivamente, los siguientes nm valores son la matriz de costos c_{ij}, a continuación nm valores de la matriz de coeficientes tecnológicos a_{ij}, los últimos m valores representan al vector del lado derecho b_j. A continuación se muestra como ejemplo la instancia de 5 agentes y 15 tareas:

5 15 17 21 22 18 24 15 20 18 19 18 16 22 24 24 16 23 16 21 16 17 16 19 25 18 21 17 15 25 17 24 16 20 16 25 24 16 17 19 19 18 20 16 17 21 24 19 19 22 22 20 16 19 17 21 19 25 23 25 25 25 18 19 15 15 21 25 16 16 23 15 22 17 19 22 24 8 15 14 23 8 16 8 25 9 17 25 15 10 8 24 15 7 23 22 11 11 12 10 17 16 7 16 10 18 22 21 20 6 22 24 10 24 9 21 14 11 14 11 19 16 20 11 8 14 9 5 6 19 19 7 6 6 13 9 18 8 13 13 13 10 20 25 16 16 17 10 10 5 12 23 36 34 38 27 33

La cual debe leerse como sigue :

$m = 5$;

$n = 15$;

$c_{ij} =$

$$\begin{pmatrix} 17 & 21 & 22 & 18 & 24 & 15 & 20 & 18 & 19 & 18 & 16 & 22 & 24 & 24 & 16 \\ 23 & 16 & 21 & 16 & 17 & 16 & 19 & 25 & 18 & 21 & 17 & 15 & 25 & 17 & 24 \\ 16 & 20 & 16 & 25 & 24 & 16 & 17 & 19 & 19 & 18 & 20 & 16 & 17 & 21 & 24 \\ 19 & 19 & 22 & 22 & 20 & 16 & 19 & 17 & 21 & 19 & 25 & 23 & 25 & 25 & 25 \\ 18 & 19 & 15 & 15 & 21 & 25 & 16 & 16 & 23 & 15 & 22 & 17 & 19 & 22 & 24 \end{pmatrix}$$

$a_{ij} =$

$$\begin{pmatrix} 8 & 15 & 14 & 23 & 8 & 16 & 8 & 25 & 9 & 17 & 25 & 15 & 10 & 8 & 24 \\ 15 & 7 & 23 & 22 & 11 & 11 & 12 & 10 & 17 & 16 & 7 & 16 & 10 & 18 & 22 \\ 21 & 20 & 6 & 22 & 24 & 10 & 24 & 9 & 21 & 14 & 11 & 14 & 11 & 19 & 16 \\ 20 & 11 & 8 & 14 & 9 & 5 & 6 & 19 & 19 & 7 & 6 & 6 & 13 & 9 & 18 \\ 8 & 13 & 13 & 13 & 10 & 20 & 25 & 16 & 16 & 17 & 10 & 10 & 5 & 12 & 23 \end{pmatrix}$$

$b_j =$

$$\begin{pmatrix} 36 \\ 34 \\ 38 \\ 27 \\ 33 \end{pmatrix}$$

En la tabla siguiente se muestran las características de las instancias empleadas en este trabajo.

Instancias utilizadas					
#	Nombre de la instancia	Tipo	m	n	Mejor conocido
1	C5100	C	5	100	1931
2	C10100	C	10	100	1402
3	C5200	C	5	200	3456
4	C10200	C	10	200	2806
5	C20200	C	20	200	2391
6	D10100	D	10	100	6349
7	D5200	D	5	200	12743
8	D10200	D	10	200	12436
9	D20200	D	20	200	12264
10	E10100	E	10	100	11577
11	E5200	E	5	200	24930
12	E10200	E	10	200	23307
13	E20200	E	20	200	22379
14	C201600	C	20	1600	18802
15	C801600	C	80	1600	16284

Cuadro 5.1: Instancias de prueba

5.2. Calibración de parámetros

Los parámetros se calibraron con un algoritmo de búsqueda armónica *(BA)*, los algoritmos hicieron cien mil llamadas a la función objetivo en cada una de las cuatro instancias, c10100, d10200, d20200, e20200.

5.2.1. Búsqueda Armónica

El algoritmo *(BA)* fue propuesto por Zong Woo Geem y Kang Seo Lee en el (2001), basado en el proceso de improvisación musical, donde los músicos buscan producir una armonía, que se representa como un vector n-dimensional ya que un en la improvisación musical, cada músico toca una nota dentro de un posible rango; de tal manera que se genera un vector armónico. Si el conjunto de notas es considerado una buena armonía, éste se guardada en la memoria de cada músico, incrementando la posibilidad de hacer una buena armonía, lo que significa que, cada variable del vector n- dimensional toma valores aleatorios dentro de un rango posible, formando un vector solución que es guardado en una la memoria armónica, [24], [16]. El *(BA)* es un algoritmo poblacional donde cada armonía representa a un individuo de la población, a continuación se describe cómo se desarrolla la heurística *(BA)*.

Algoritmo 13 BA

Introducir los datos de entrada: Número de parámetros a calibrar, rango de cada uno de los parámetros y el conjunto de instancias de prueba
Introducción de los parámetros requeridos por la heurística HMR= PAR= b=
El algoritmo genera una población inicial de individuos
Se evalúan los individuos
while Mientras no se cumpla el criterio de paro **do**
    Se genera un nuevo individuo
    Si el nuevo individuo es estadísticamente mejor que el peor individuo de la población, el nuevo individuo lo sustituye
end while
Se muestra la mejor solución obtenida

Los parámetros que se obtuvieron para las técnicas desarrolladas *AL*, *ABG* y *MCM* se muestran en las siguiente tabla.

AL	
población	13
evaluaciones	100000
alfa	0.12205
beta	0.060248
P_g	0.21493
P_a	0.58673
V1	0.1

ABG	
población	3
evaluaciones	100000
G	0.642268
beta	0.89416
P_g	0.3364
P_a	0.5236

MCM	
NC	3
NS	3
evaluaciones	100000
P_g	0.4410
P_a	0.7154
fcla	0.9110
ifg	0.02297
cfg	0.2577

Cuadro 5.2: Parámetros calibrados

Se realizaron los experimentos correspondiente comparando el rendimiento de las técnicas propuestas entre ellas y con algunas existentes en el estado del arte, los datos comparativos fueron tomados de las tabla que se muestra continuación.

Comparativas Mejores Resultados	
R& A	[27]
HRL	[27]
HA-N	[27],[49]
VBP-i	[76]
VBP-j	[76], [3]
BLM	[76]
VBP-Y	[76]
BVP-RA	[76]
BT-L	[76]
BT-NI	[76]
AG-CB	[76]
BT	[19]
BTCE	[19]
AG	[19], [76], [13]
AGCA	[40]
R& C & P	[53]
RRB	[30]
AC	[30]
HRRBAC	[30]
MRPAG	[45]

Comarativas Mejores Resultados	
VBP	[4]
HPAG	[4]
HH	[4]
VBPY	[2]
BTY	[2]
AG	[2], [19], [13]
BTCE	[2], [19], [74]
TS	[2], [19]
RT	[2]
BTRA	[73]
EXpress-MP	[73]

Cuadro 5.3: Técnicas utilizadas para comparar el estado del arte.

5.3. Desarollo experimental

Se realizaron veinte experimentos con los parámetros obtenidos anteriormente, con cada una de las instancias de prueba para cada una de las técnicas propuestas; los resultados obtenidos son analizados con la técnica de remuestreo Bootstrab, el cual es utilizado para aproximar la varianza de un análisis estadístico y obtener intervalos de confianza. Estos datos ayudan a fundamentar una hipótesis sobre parámetros de interés.

Se emplea una prueba no paramétrica para comparar el rango medio de dos muestras relacionadas, en este caso se ocupa la prueba de Wilcoxon o conocida como prueba de los signos, para determinar si existen diferencia entre los algoritmos desarrollados y los existentes en el estado del arte, como comparativo para evaluar el desempeño de las técnicas desarrolladas.

Los algoritmos fueron desarrollados en MATLAB R2015a y probados en una computadora portátil Lenovo X220 con un procesador Intel(R) Core(TM) i5-2520M CPU @ 2.50 GHz, con una memoria RAM de 8GB y con sistema operativo Windows 10.

Capítulo 6

Experimentación

En este capítulo, se presentan los resultados obtenidos por la implementación de las técnicas propuestas.

En la primera sección, se caracterizan los comportamientos de las técnicas desarrolladas y probadas con las instancias antes mencionadas. Como se dijo en la metodología se realizaron 20 experimentos independientes de cada técnica sobre cada una de las instancias. En las tablas 6.1, 6.3, 6.5, 6.7, 6.9, 6.11, 6.13, 6.15, se muestran los mejores valores obtenidos, los peores, el pro medio, la mediana, la varianza y la desviación estándar, para cada una de las instancias probadas.

Por otro lado en las tablas 6.2, 6.4, 6.6, 6.8, 6.10, 6.12, 6.14, 6.16, se presenta una prueba de boobstrap por cada una de las técnicas desarrolladas y los datos obtenidos de las instancias probadas, mostrando los intervalos de confianza para la mediana y la varianza.

AL1						
INSTANCIA	MEJOR	PROMEDIO	PEOR	MEDIANA	VARIANZA	DES. EST.
C5100	1952	2013.15	2070	2015	1101.18684	33.1841354
C10100	1439	1465.05	1499	1463	343.628947	18.5372314
C5200	3629	3698.6	3758	3708.5	1154.98947	33.9851361
C10200	2845	2874.6	2907	2872	353.2	18.7936159
C20200	2458	2482.35	2503	2484.5	188.239474	13.7200391
D10100	6495	6535.45	6575	6534.5	489.839474	22.1323174
D5200	12838	13217.15	13327	13275.5	26299.7132	162.171863
D10200	12588	12630.6	12703	12625	682.357895	26.1219811
D20200	12626	12681.4	12761	12676.5	1068.98947	32.6954045
E10100	11979	12149.85	12430	12122	16083.1868	126.819505
E5200	25239	26195.1	27994	25873	1100698.83	1049.14195
E10200	23841	24045.95	24394	24013.5	21644.6816	147.121316
E20200	23181	23679.3	24506	23649.5	105607.379	324.97289
C201600	18970	18999.85	19067	18995	515.292105	22.7000464
C801600	17140	17323.85	17543	17334	8971.08158	94.7157937

Cuadro 6.1: AL1 Resultados

El algoritmo *AL1* muestra un comportamiento estable en la obtención de sus soluciones, en los mejores casos presenta un error con respecto al mejor reportado menor al 1 % (D5200 y C201600) y en el peor de los casos un error de 5 % (C5200 y C801600);

PRUEBA BOOBSTRAP AL1				
INSTANCIAS	MEDIANA LI	MEDIANA LS	VARIANZA LI	VARIANAZA LS
C5100	542.905263	1641.41842	2005	2024.5
C10100	178.934211	479.484211	1453.5	1475
C5200	540	1778.43158	3679.5	3714
C10200	195.207895	495.881579	2864.5	2888
C20200	111.789474	257.881579	2471.5	2492
D10100	217.736842	714.05	6523.5	6545
D5200	474.239474	44925.1053	13265	13283.5
D10200	244.147368	1235.14737	12613.5	12640.5
D20200	474.736842	1752.47105	12663.5	12700
E10100	6065.46053	25161.5237	12072.5	12175
E5200	565921.053	1473880.91	25355.5	26465
E10200	7393.62895	36059.5895	23951	24104
E20200	36596.9579	177672.155	23460	23791.5
C201600	107.368421	981.042105	18986	19001.5
C801600	3858.32632	14858.3684	17270	17370

Cuadro 6.2: PRUEBA DE BOOBSTRAP AL1

AL2						
INSTANCIAS	MEJOR	PROMEDIO	PEOR	MEDIANA	VARIANZA	DES. EST.
C5100	1964	2024.45	2070	2031.5	896.576316	29.94288423
C10100	1455	1474.45	1498	1474	160.576316	12.67187105
C5200	3477	3696.75	3757	3709.5	3653.56579	60.44473335
C10200	2839	2874.55	2904	2879.5	310.05	17.60823671
C20200	2460	2490.55	2526	2489	310.05	17.60823671
D10100	6489	6534.6	6582	6539.5	544.147368	23.32696655
D5200	12830	13205.55	13308	13276	24459.8395	156.3964177
D10200	12592	12632.55	12667	12636	507.628947	22.53062244
D20200	12614	12693.85	12755	12693.5	1083.6079	32.91820005
E10100	11975	12134.85	12291	12152.5	10523.3974	102.5836116
E5200	25059	25634.2	27893	25428	620808.484	787.9140081
E10200	23836	24096.25	24457	24097.5	32946.3026	181.5111639
E20200	23189	23806.6	24217	23820	52552.8842	229.2441585
C201600	18979	19013.9	19061	19010	609.357895	24.68517561
C801600	17366	17593.65	18157	17569	32448.029	180.1333643

Cuadro 6.3: AL2 Resultados

Los resultados mostrados en la tabla del algoritmo *AL2* en los mejores casos presenta un error con respecto al mejor reportado menor al 1 % en (C5200, D5200, E5200 y C201600) y el peor de los casos el error presentado es del 6.6 % lo presenta para la instancia (C801600).

PRUEBA BOOBSTRAP AL2				
INSTANCIAS	MEDIANA LI	MEDIANA LS	VARIANZA LI	VARIANAZA LS
C5100	440.8	1338.19737	2010	2043.5
C10100	83.7131579	233.292105	1465	1480
C5200	552.092105	8570.68158	3689	3726.5
C10200	134.365789	482.210526	2871	2883.5
C20200	132.894737	486.736842	2481.5	2496
D10100	243.313158	866.526316	6520.5	6547.5
D5200	879.726316	41988.9974	13233	13282
D10200	265.831579	710.155263	12620.5	12648
D20200	433.157895	1752.88421	12680	12715
E10100	6490.43158	14085.9447	12043	12215.5
E5200	33087.3263	1259905.43	25242	25527
E10200	16838.5763	48752.6605	23947	24197.5
E20200	19272.5132	95498.9342	23710.5	23915
C201600	325.502632	860.134211	18999	19029.5
C801600	10112.8921	65521.8395	17495	17662.5

Cuadro 6.4: PRUEBA DE BOOBSTRAP AL2

AL3						
INSTANCIAS	MEJOR	PROMEDIO	PEOR	MEDIANA	VARIANZA	DES. EST.
C5100	1949	1974.1	1998	1974	110.410526	10.50764133
C10100	1426	1450.3	1474	1453	128.221053	11.32347352
C5200	3562	3588.7	3617	3591.5	234.11579	15.30084277
C10200	2845	2864.9	2889	2863	154.51579	12.43043802
C20200	2447	2472.2	2500	2471	223.326316	14.94410639
D10100	6462	6516.15	6560	6517.5	592.028947	24.33164498
D5200	12829	13127.25	13257	13184.5	21633.25	147.0824599
D10200	12560	12633.65	12696	12631.5	1000.45	31.62989093
D20200	12602	12669.6	12730	12667.5	1041.72632	32.27578529
E10100	11854	11991.35	12280	11963	11689.7132	108.1189769
E5200	24983	25628.3	26930	25359	403053.59	634.8650167
E10200	23882	24092.4	24569	24064	27589.6211	166.1012374
E20200	23273	23848.7	24510	23901	132409.59	363.8812849
C201600	18985	19030.45	19089	19025.5	866.260526	29.43230413
C801600	19568	19708.4	19919	19701.5	8557.51579	92.50684185

Cuadro 6.5: AL3 Resultados

En el caso del *AL3* los resultados en el mejor de los casos con respecto de la mejor solución reportada presenta un error del 1 % para las instancias (C5100, D5200, D10200, E5200 y C201600) y en el peor de los casos se presenta un error del 20.2 % para la instancia (C801600).

PRUEBA BOOBSTRAP AL3				
INSTANCIAS	MEDIANA LI	MEDIANA LS	VARIANZA LI	VARIANAZA LS
C5100	37.0105263	196.997368	1971	1978.5
C10100	52.5131579	214.431579	1446	1455
C5200	116.778947	339.147368	3578	3596
C10200	78.1342105	230.2	2860.5	2871.5
C20200	112.421053	334.576316	2467	2480.5
D10100	260.892105	950.831579	6503	6526
D5200	6573.25	32593.6289	13161.5	13202.5
D10200	334.407895	1745.35789	12625.5	12644
D20200	455.936842	1632.66053	12663.5	12686
E10100	4908.2	19991.0105	11925	12050.5
E5200	174021.945	583121.818	25183.5	25930
E10200	9118.46316	51645.0947	24011	24137
E20200	73382.0421	192957.589	23605	24026
C201600	387.776316	1309.08158	19016	19040
C801600	3942.35789	13240.6842	19656.5	19752

Cuadro 6.6: PRUEBA BOOBSTRAP AL3

AL4						
INSTANCIAS	MEJOR	PROMEDIO	PEOR	MEDIANA	VARIANZA	DES. EST.
C5100	1971	2024.55	2047	2029.5	411.102632	20.275666
C10100	1437	1469.1	1503	1469.5	410.936842	20.2715772
C5200	3661	3722.9	3762	3729	774.305263	27.82634117
C10200	2844	2885.35	2932	2883.5	396.134211	19.90312062
C20200	2449	2481	2519	2482	347.052632	18.62934866
D10100	6498	6530.8	6571	6533.5	374.8	19.35975206
D5200	12809	13236.95	13321	13280	21688.3658	147.2697042
D10200	12571	12630.7	12684	12628.5	870.536842	29.50486133
D20200	12597	12680.1	12744	12680.5	1020.51579	31.94551282
E10100	11939	12162.35	12365	12191	20960.2395	144.7765156
E5200	25172	26231.1	28184	25613.5	1174477.46	1083.733114
E10200	23741	24023.45	24282	24020	23218.05	152.3747026
E20200	23185	23621.6	23945	23625.5	44707.8316	211.4422654
C201600	18946	19005.65	19079	19007	1105.92368	33.2554309
C801600	16842	17140.85	17662	17087	49015.0816	221.3934994

Cuadro 6.7: AL4 Resultados

Los resultados obtenidos por *AL4* las instancias que presentan un error del 1 % con respecto de la mejor solución reportada son (D5200, D10200, E5200 y C201600) y el peor de los casos el errores de 5.9 % para la instancia (C5200).

PRUEBA BOOBSTRAP AL4				
INSTANCIAS	MEDIANA LI	MEDIANA LS	VARIANZA LI	VARIANAZA LS
C5100	130.197368	700.555263	2023	2036
C10100	215.884211	574.410526	1459	1482
C5200	342.621053	1193.27368	3713	3735
C10200	125.081579	596.028947	2877	2894
C20200	164.273684	507.607895	2471.5	2490.5
D10100	153.986842	531.102632	6517	6539.5
D5200	533.989474	38631.6711	13266	13304
D10200	424.365789	1347.83947	12613	12648
D20200	321.207895	1541.92368	12665	12692
E10100	12194.2211	28497.9579	12071	12253
E5200	620106.011	1487431.25	25407	27033
E10200	11818.9368	33647.0816	23929.5	24109
E20200	20930.2395	62886.6605	23530	23791
C201600	429.144737	1776.04211	18993	19014
C801600	19880.0632	75777.3132	17011.5	17236.5

Cuadro 6.8: PRUEBA BOOBSTRAP AL4

AL5						
INSTANCIAS	MEJOR	PROMEDIO	PEOR	MEDIANA	VARIANZA	DES. EST.
C5100	1980	2028.2	2064	2027	500.168421	22.36444547
C10100	1437	1468.7	1506	1467	295.905263	17.20189708
C5200	3605	3712.6	3765	3711.5	1288.67368	35.8981014
C10200	2844	2874.75	2917	2874.5	374.197368	19.34418177
C20200	2463	2486.1	2515	2484.5	166.305263	12.89593979
D10100	6491	6540.8	6564	6544	397.642105	19.9409655
D5200	12832	13225.6	13327	13272.5	19199.6211	138.5626972
D10200	12572	12633.1	12690	12637	775.252632	27.84335884
D20200	12645	12701	12781	12698.5	1161.47368	34.08040029
E10100	11948	12187.05	12494	12167.5	15941.8395	126.2609974
E5200	25132	26089.05	27905	25481	1161600.26	1077.775608
E10200	23854	24156.75	24438	24128.5	31505.4605	177.4977761
E20200	23435	23832.7	24266	23775.5	70483.2737	265.4868616
C201600	18967	19010.05	19059	19008.5	516.786842	22.73294618
C801600	17038	17307.7	17519	17275.5	23553.5895	153.4717872

Cuadro 6.9: AL5 resultados

Los resultados obtenidos con la técnica *AL5* muestran un error con respecto de la mejor solución reportada en el mejor de los casos del 1 % para las instancias (D5200, E5200 y C201600) mientras que en el peor de los casos el error es un poco menor al 5 % para las instancias (E20200 y C801600).

PRUEBA BOOBSTRAP AL5				
INSTANCIAS	MEDIANA LI	MEDIANA LS	VARIANZA LI	VARIANAZA LS
C5100	258.884211	771.168421	2013.5	2040.5
C10100	116.828947	460.471053	1463	1472
C5200	419.628947	2514.58947	3699.5	3733.5
C10200	154.618421	563.989474	2868.5	2883.5
C20200	66.4631579	266.957895	2481.5	2492
D10100	191.207895	650.063158	6529.5	6556
D5200	1051.22105	37343.4211	13249	13292
D10200	370.660526	1249.90526	12617	12650.5
D20200	480.723684	1921.46316	12683	12717
E10100	6168.05263	27108.2105	12139	12238
E5200	548706.779	1493520.05	25313	26630
E10200	17674.2395	44577.6079	24036.5	24256
E20200	37005.9579	95471.8184	23625	23995
C201600	204.618421	859.168421	19003	19016
C801600	14351.1868	32132.8711	17226.5	17450

Cuadro 6.10: [PRUEBA BOOBSTRAP AL5

AL6						
INSTANCIAS	MEJOR	PROMEDIO	PEOR	MEDIANA	VARIANZA	DES. EST.
C5100	1957	1973.7	1997	1974.5	110.957895	10.53365534
C10100	1424	1450.5	1471	1452	157.421053	12.54675467
C5200	3560	3592.65	3643	3593	353.186842	18.79326587
C10200	2833	2865.6	2892	2865	257.831579	16.05713483
C20200	2449	2472.1	2491	2474	158.094737	12.57357295
D10100	6486	6525.05	6562	6522.5	484.997368	22.0226558
D5200	12792	13099.15	13254	13193	30053.5026	173.3594608
D10200	12594	12632.65	12705	12629	809.292105	28.44805978
D20200	12633	12673.95	12713	12682.5	646.681579	25.4299347
E10100	11802	12012.6	12213	12009	9742.04211	98.7017837
E5200	25015	25662.6	27168	25253	487935.2	698.5235859
E10200	23699	24251.2	24969	24198.5	109029.326	330.1958908
E20200	23569	23916.9	24178	23943.5	34253.9895	185.0783333
C201600	18988	19030.05	19081	19029.5	682.892105	26.13220437
C801600	18749	19116.4	19364	19186	39050.5684	197.6121667

Cuadro 6.11: AL6 Resultados

Las soluciones obtenidas por la técnica *AL6* presentan un error ene el mejor de los casos del 1 % con respecto de la mejor solución reportada para las instancias (C10200, D5200, E5200, C201600)y ene l peor de los casos del 15 % para la instancia (C801600).

PRUEBA BOOBSTRAP AL6				
INSTANCIAS	MEDIANA LI	MEDIANA LS	VARIANZA LI	VARIANAZA LS
C5100	49.5026316	176.197368	1967.5	1977.5
C10100	82.8947368	240.344737	1441.5	1458
C5200	116.786842	520.568421	3585	3600
C10200	115.628947	376.684211	2856	2873
C20200	74.9342105	227.313158	2468	2480.5
D10100	241.368421	692.344737	6513.5	6540.5
D5200	12733.6842	37928.6816	13040	13219
D10200	332.905263	1325.48421	12618.5	12643.5
D20200	338.765789	877.010526	12656	12688
E10100	4007.05263	15951.8	11968.5	12053.5
E5200	153112.661	702094.905	25154	25761.5
E10200	49876.2395	168831.274	24080	24446.5
E20200	18043.7342	48098.5684	23821.5	24055
C201600	324.344737	920.536842	19015	19041
C801600	18762.6816	52956.4316	19006	19273.5

Cuadro 6.12: PRUEBA BOOBSTRAP AL6

ABG						
INSTANCIAS	MEJOR	PROMEDIO	PEOR	MEDIANA	VARIANZA	DES. EST.
C5100	1957	1977.6	2012	1976.5	194.147368	13.93367749
C10100	1443	1482.75	1518	1484.5	461.039474	21.47182977
C5200	3483	3524.3	3551	3526.5	389.589474	19.73802102
C10200	2860	2909.7	2954	2912.5	634.431579	25.18792526
C20200	2517	2575.3	2779	2566	3253.69474	57.04116704
D10100	6499	6586.05	6646	6585	1620.36579	40.25376739
D5200	12805	12872.05	12913	12870	763.523684	27.63193233
D10200	12620	12730.5	13025	12718	7286.68421	85.36207712
D20200	12689	12808.4	12952	12805	4596.77895	67.79954976
E10100	12158	12705.05	13413	12703.5	146013.524	382.1171596
E5200	25450	25962.25	27404	25753.5	369100.092	607.536083
E10200	24137	25595.75	28786	25335.5	1180420.3	1086.471492
E20200	24696	25487.85	26504	25431	289457.397	538.0124509
C201600	19131	19540.7	23709	19343.5	968422.221	984.0844583
C801600	20499	23558.65	31251	20845.5	18480899.4	4298.94166

Cuadro 6.13: ABG Resultados

Para la técnica *ABG* los resultados obtenidos un buen comportamiento los mejores casos tienen un error del 1 % para las instancias (C5200 y D5200) el peor de los casos lo presenta la instancia (C801600) con un erro del 26 %.

PRUEBA BOOBSTRAP ABG				
INSTANCIAS	MEDIANA LI	MEDIANA LS	VARIANZA LI	VARIANAZA LS
C5100	73.0105263	326.989474	1970	1983
C10100	264.315789	652.410526	1468	1498.5
C5200	188.660526	572.365789	3515	3538.5
C10200	278.576316	997.713158	2898	2924.5
C20200	568.042105	7502.78684	2546.5	2581
D10100	849.081579	2328.51579	6558.5	6617
D5200	341.039474	1230.05263	12859	12889
D10200	1481.81842	16236.2395	12691.5	12747
D20200	2188.45	7194.58947	12761.5	12846.5
E10100	79853.2632	209467.987	12460	12944.5
E5200	40964.8526	624194.682	25616	25861
E10200	122925.905	2320909.26	25135	25585
E20200	145046.695	411094.155	25157.5	25709.5
C201600	3097.94474	2615329.22	19278	19367.5
C801600	8758673.47	23893955.9	20821	25044.5

Cuadro 6.14: PRUEBA BOOBSTRAP ABG

MCM						
INSTANCIAS	MEJOR	PROMEDIO	PEOR	MEDIANA	VARIANZA	DES. EST.
C5100	1952	1975.45	2011	1973.5	232.471053	15.2470014
C10100	1442	1456.8	1477	1453.5	107.957895	10.3902789
C5200	3528	3565.75	3608	3567	480.618421	21.9230112
C10200	2878	2933.1	2966	2936.5	552.410526	23.5034152
C20200	2472	2495.05	2519	2494.5	158.471053	12.5885286
D10100	6640	6766.3	6850	6770	2568.22105	50.6776189
D5200	13110	13209.6	13310	13215	3242.56842	56.9435547
D10200	13134	13346	13447	13355.5	6383.1579	79.8946675
D20200	13339	13560.4	13968	13533	22825.2	151.080111
E10100	12703	13485.55	14024	13499	107288.366	327.549028
E5200	25092	27283.95	28510	27383	639660.261	799.787635
E10200	27366	28218.05	29478	28017	474540.892	688.869285
E20200	27643	29218.9	30467	29351	725894.516	851.994434
C201600	22792	24307.45	24742	24351	170165.313	412.510986
C801600	20087	20309.3	20769	20304.5	22238.8526	149.126968

Cuadro 6.15: MCM Resultados

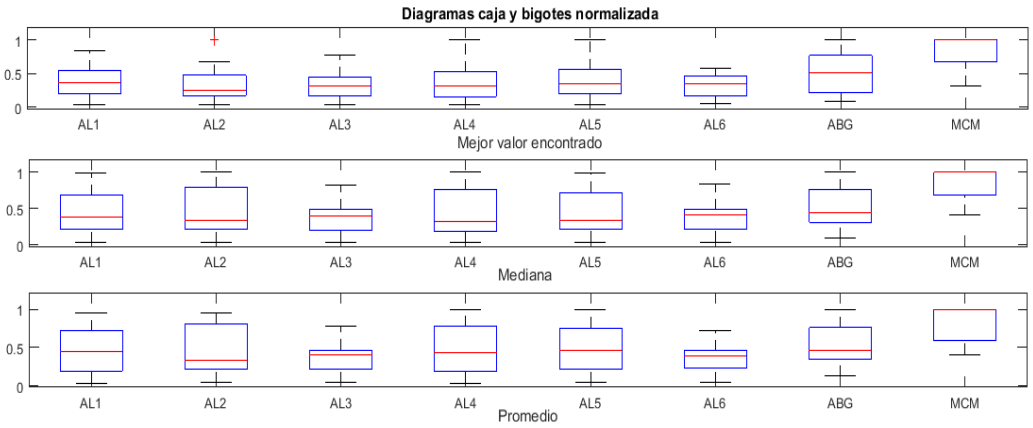
La técnica *MCM* presentan un comportamiento poco estable ya que en el mejor de los casos presenta un error con respecto al mejor reportado del 1 % para las instancias (C5100 y E5200) y en el peor de los casos el error presentado del 24 % (E20200 y C801600).

PRUEBA BOOBSTRAP MCM				
INSTANCIAS	MEDIANA LI	MEDIANA LS	VARIANZA LI	VARIANAZA LS
C5100	92.7263158	370.344737	1966	1981
C10100	48.2394737	143.2	1449.5	1461.5
C5200	235.628947	689.839474	3551	3576.5
C10200	270.260526	873.905263	2920	2949
C20200	77.4842105	213.607895	2487	2502.5
D10100	1071.08158	4292.89211	6748.5	6790
D5200	1697.98684	4804.93421	13180	13245.5
D10200	2710.41053	11010.4737	13310.5	13390
D20200	7646.89211	41219.1579	13482.5	13608
E10100	51726.6211	170001.208	13278.5	13730
E5200	215170.513	1127577.92	26955	27809.5
E10200	230371.945	662817	27768.5	28606
E20200	321685.418	1031588.05	28867	29729
C201600	20580.3158	405623.516	24311	24473
C801600	6737.37895	43750.6605	20224	20348

Cuadro 6.16: PRUEBA DE BOOBSTRAP MCM

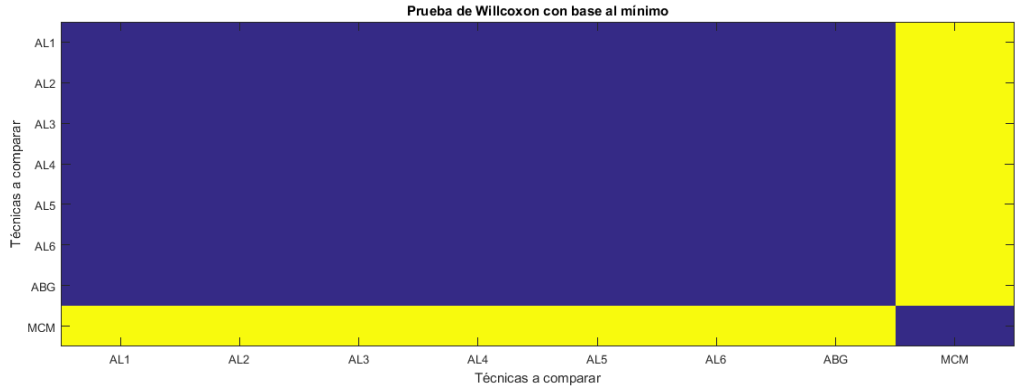
Con base en los resultados mostrados en las tablas anteriores se puede observar que las técnicas que presentan los mejores resultados son *AL1* y *AL2* ya que presentan mayor estabilidad, ya que el error con respecto de la mejor solución reportada oscila entre 1 % y 5 % y las que presentan el peor comportamiento debido a su margen de error son *AL3*, *ABG* y *MCM* ya que su margen de error varían entre el 1 % y el 20 %.

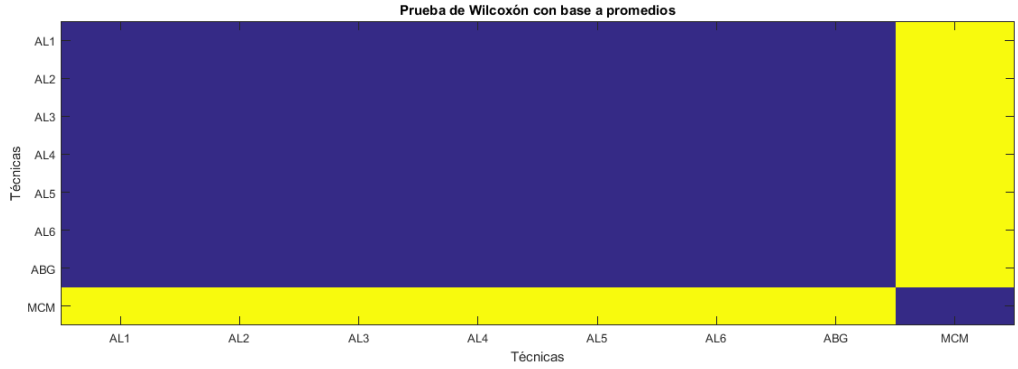
La siguiente gráfica se representa con diagramas de caja y bigotes, el comportamiento de las técnicas desarrolladas, para los mejores valores obtenidos, con base a la medianas y a los valores promedio.



Como se puede observar el *MCM* es el que presenta el peor comportamiento seguido del *ABG*, también se puede observar que las técnicas de luciérnagas son las que presentan los mejores resultados sin embargo *(AL3)* y *(AL6)* muestran un comportamiento más estable y una menor dispersión es su resultados.

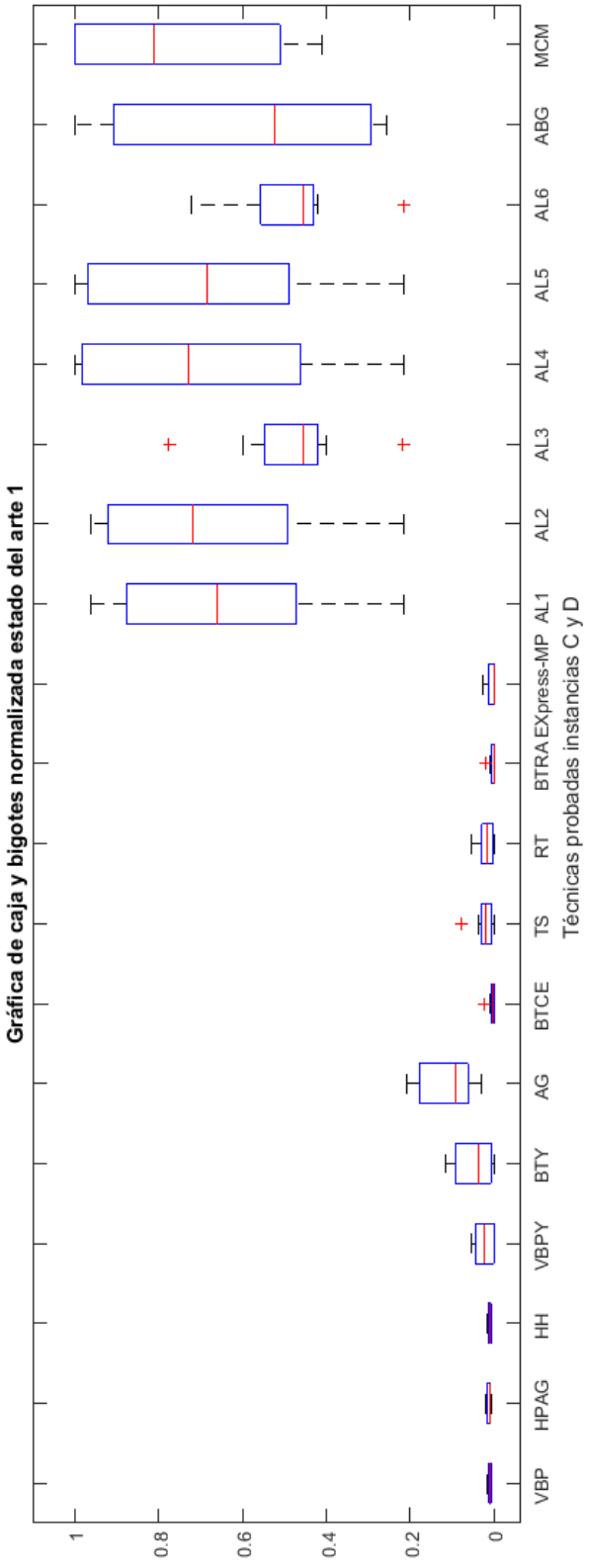
Con el objeto de comprobar estas aseveraciones se realizó la prueba de Wilcoxon.



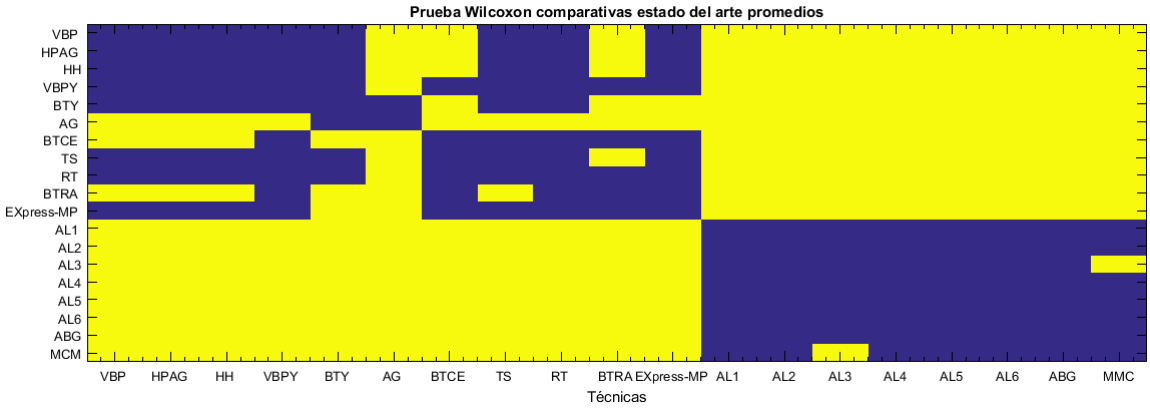


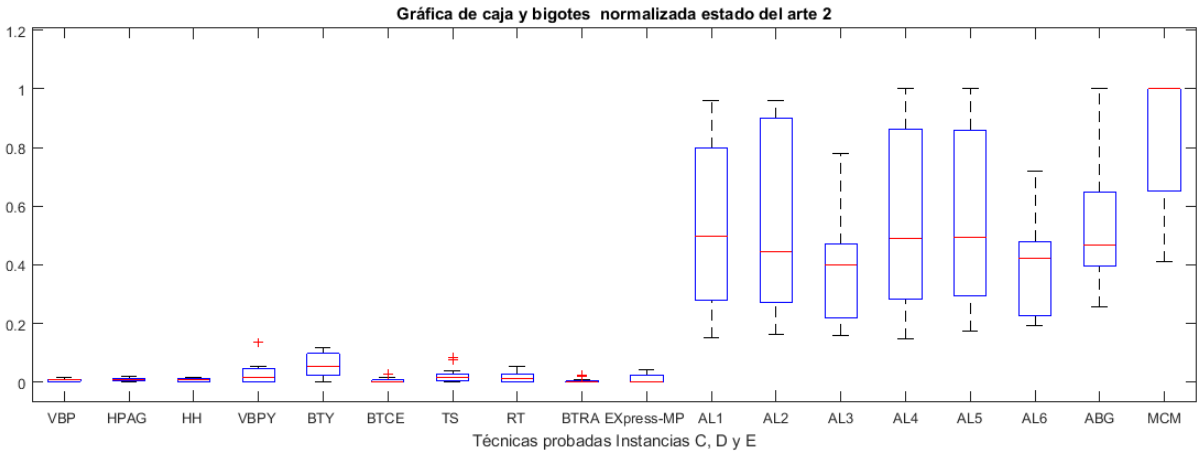
Con base en las pruebas de Wilcoxon de se puede resumir que el peor comportamiento es presentado por la técnica *MCM*, es estadísticamente diferente a las demás y con base a la gráficas de caja y bigotes es inferior a las demás.

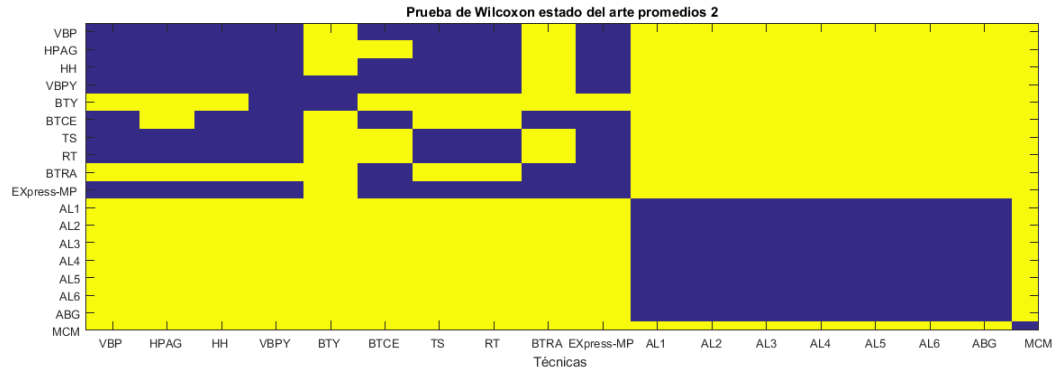
Las siguientes tablas muestran un comparativo de las técnicas propuestas contra las reportadas en el estado del arte.

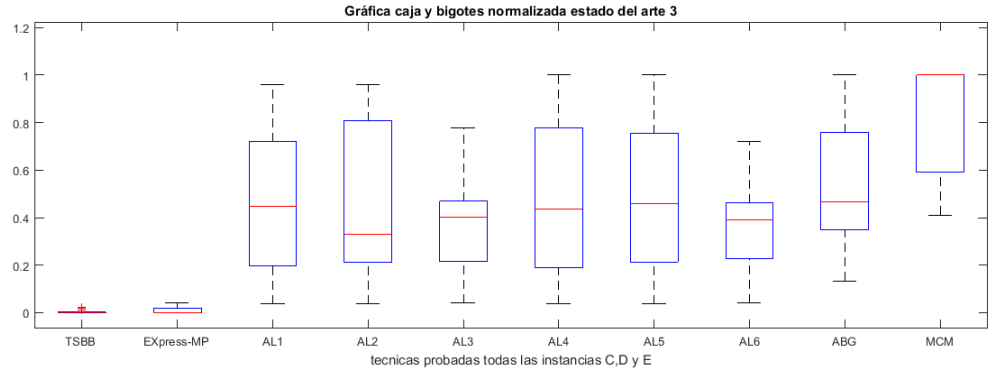

Gráfica de caja y bigotes normalizada estado del arte 1
1
0.8
0.6
0.4
0.2
0
VBP
HPAG
HH
VBPY
BTY
AG
BTCE
TS
RT
BTRA EXpress-MP
AL1
AL2
AL3
AL4
AL5
AL6
ABG
MCM
Técnicas probadas instancias C y D

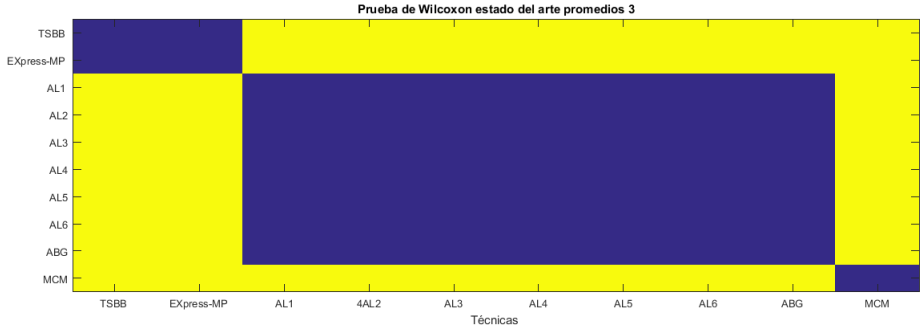
Como se puede observar en la tabla anterior las técnicas del estado del arte que presentan el mejor comportamiento son: búsqueda tabú con cadena de expulsión y ramificación y acotamiento, por otro lado, las técnicas desarrolladas en este trabajo los mejores resultados son obtenidos por *AL3* y *AL6* mientras que el *MCM* presenta el peor comportamiento presentado para las instancias probadas.

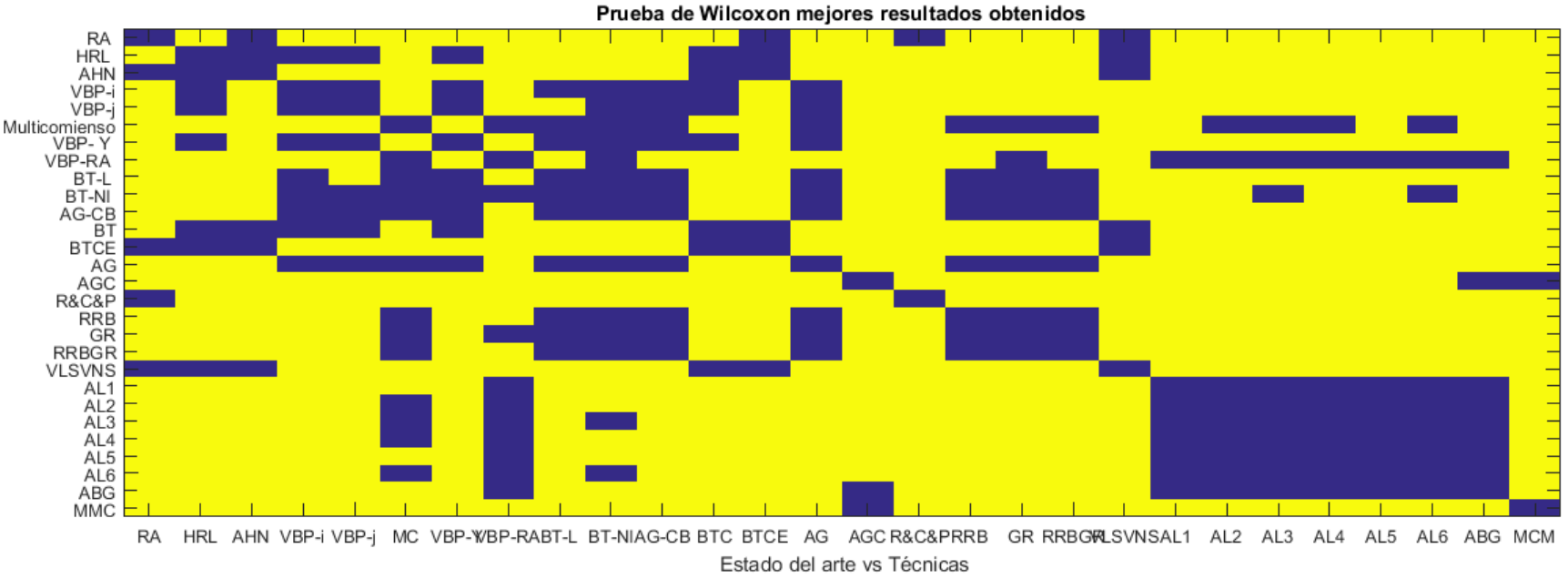





Prueba de Wilcoxon estado del arte promedios 2
VBP
HPAG
HH
VBPY
BTY
BTCE
TS
RT
BTRA
EXpress-MP
AL1
AL2
AL3
AL4
AL5
AL6
ABG
MCM
Técnicas


Gráfica caja y bigotes normalizada estado del arte 3
1.2
1
0.8
0.6
0.4
0.2
0
TSBB
EXpress-MP
AL1
AL2
AL3
AL4
AL5
AL6
ABG
MCM
tecnicas probadas todas las instancias C,D y E


Prueba de Wilcoxon estado del arte promedios 3
TSBB
EXpress-MP
AL1
AL2
AL3
AL4
AL5
AL6
ABG
MCM
TSBB
EXpress-MP
AL1
4AL2
AL3
AL4
AL5
AL6
ABG
MCM
Técnicas


Prueba de Wilcoxon mejores resultados obtenidos
RA
HRL
AHN
VBP-i
VBP-j
Multicomienso
VBP- Y
VBP-RA
BT-L
BT-NI
AG-CB
BT
BTCE
AG
AGC
R&C&P
RRB
GR
RRBGR
VLSVNS
AL1
AL2
AL3
AL4
AL5
AL6
ABG
MMC
RA HRL AHN VBP-i VBP-j MC VBP-Y VBP-RA BT-L BT-NI AG-CB BTC BTCE AG AGC R&C&P RRB GR RRBGR VLSVNS AL1 AL2 AL3 AL4 AL5 AL6 ABG MCM
Estado del arte vs Técnicas

Como se puede observar en los comparativos anteriores, las técnicas propuesta *AL* en sus seis versiones, *ABG* y *MCM*, son estadísticamente similares en comportamiento con algunas técnicas mostradas en el estado del arte, Donde *(VBP-RA)* es similar a las seis versiones de *AL* y *ABG*, *(BML)* es similar a *AL2*, *AL3*, *AL4* y *AL6*, mientra que *(BT-NI)* es similar a *AL3* y *AL6* y para terminar el *(AGC)* es similar al *MCM*.

Capítulo 7

Conclusiones

En el presente trabajo se caracterizaron tres técnicas pertenecientes a la ***IP***, *AL*, *ABG* y *MCM*, las cuales fueron adaptadas al *PAG*, desarrollando seis versiones para el *AL*, una para el *ABG* y otra para el *MCM*, a diferencia de los algoritmos originales que parten de soluciones iniciales aleatorias, en este trabajo se buscó partir de soluciones iniciales generadas en tres etapas, la primera por un algoritmo glotón, la segunda por la relajación del *PL* y la tercera de manera aleatoria, con la idea de partir de soluciones de mejor calidad, sin perder diversidad.

Los experimentos realizados con las técnicas propuestas presentan un comportamiento similar entre las seis versiones del *AL* y el *ABG*; mientras que la técnica *MCM* resultó ser distinta y ofreció un menor rendimiento comprada con el resto.

Las técnicas desarrolladas fueron probadas mostrando que los algoritmos desarrollados pueden entregan soluciones de buena calidad; estos resultados fueron comparadas con algunas técnicas de las ya existentes en la literatura, aunque su rendimiento quedó por debajo; se encontraron similitudes con algunas de ellas, entre las cuales aparecen una técnica búsqueda tabú y una técnica de variable de búsqueda profunda *(VBP)*.

Una de las ventajas es que los algoritmos desarrollados alcanzan soluciones de buena calidad cien mil llamadas a la función objetivo, el hecho de que sean algoritmos poblacionales permite observar que mientras los individuos cambian conforme pasa el tiempo, permite la aparición de rasgos predominantes en la población los cuales pueden emplearse para guiar las soluciones.

En los años recientes técnicas pertenecientes a la *IP* como son el *AL* y el

ABG han tomado fuerza para resolver problemas como ruteo de vehículos donde las adaptaciones incurren en buenos resultados, como se mencionó el *PAG* es un subproblema del ruteo de vehículos, razón por la cual en este trabajo se decidió su implementación, sin embargo, debe hacerse un estudio amplio en un problema en particular ya que se puede suponer que mejorará la solución de un problema complejo mejorando las soluciones de su subproblema. Debido a que no habían sido reportadas para el *PAG*, pueden servir de punto de referencia para futuras investigaciones y mejoramiento de los elementos aquí propuestos.

Una de las desventajas de las técnicas implementadas es que no ofrecen la mejor solución reportada hasta el momento, pero cabe mencionar que la complejidad computacional de los algoritmos propuestos son del orden $O(n^2)$ que comparado con algunas de les técnicas que ofrecen los mejores resultados como *(R & A)* son de orden exponencial razón por la cual; se presentan como líneas a seguir en investigaciones futuras la hibridación de las técnicas con búsqueda tabú y variables de búsquedas profundas, ya que éstas se reportan en la literatura como aquellas con la mejor capacidad para resolver el problema. Así mismo se propone otro tipo de codificaciones para mejorar el comportamiento de las técnicas desarrolladas como trabajo a futuro.

En base a lo resultados obtenidos la técnica en la que debería de priorizarse su estudio seria el *AL*. Debido a que se observo, que los resultados para las instancias de tipo *D* y para las instancias de larga escala como son las C201600 y C801600 a pesar de que son poco estudiadas en la literatura presentan errores de aproximadamente el 3 % sobre la mejor solución encontrada.

Apéndice A

Anexo I: Glosario

En este se muestra el significado de las abreviaturas mostradas en el estado del arte.

Abreviatura y significado en español		Como se encuentran en el estado del Arte	
PAG	Problema de asignación generalizada	GAP	Generalized assignment problem
(R y A)	Ramificación y Acotamiento	(B & B)	Branch and Bound
(RL)	Relajación lagrangiana	(LR)	Lagrangian Relaxation
(RS)	Relajacion sustitutiva	(SR)	Surrogate Relaxation
(PL)	Prorama Lineal	(LP)	Linear program
NLPAG	PAG No Linelal	NLGAP	Non linear GAP
VBP	Variable de búsqueda profunda	VDSH	Variable de busqueda profunda
MMSH	Max-Min Sistema de Hormigas	MMAS	Max-Min Ant System
HH	Hibrido Heurístico	HH	hybrid heuristic
HPAG	Heuristico del Problema de Asignación Generalizada	HGAP	Heuristic GAP
RS	Recocido Simulado	SA	Simulated annealing
BT	Búsqueda Tabú	TS	Tabu Search
AG	Algoritmo Genético	GA	Genetic Algorithm
R & P	Ramificación y precio	B & P	Branch and price
RT	Reencadenamiento de trayectorias	PR	Path Relinking
R & C	Ramificación y corte	B & C	Branch and Cut
PBCAA	Procedimientos de Búsqueda Codiciosa Aleatorizada Adaptativa	GRASP	Greedy Randomized Adaptative Search procedures
CE	Cadenas de Expulsión	EC	Ejection Chains
CEPM	Cadenas de Expulsión paralelas Multi-comienzo	MPEC	Multistart Parallel Ejection Chain
CEPC	Cadenas de expulsión paralelas cooperativas	CPEC	Coooperative pararllel Ejection Chains
MS	Método de subgradiente	SM	Subgradient method
R & C & P	Ramimificación y corte y precio	B & C & P	branch-and-cut-and-price
EPAG	Estocastico PAG	SGAP	stocastic GAP
PRDC	Primero Rama Despues Corte	BFCS	Branch first cut second
PCDR	Primero Corte Despues Rama	CFBS	cut first branch second
RCS	Rama y Corte Simultaneo	BCS	Branch and cut simultaneous
RTCE	Reencadenamiento de trayectorias con cadenas de expulsión	PREC	Path Relinking Ejection Chains

Cuadro A.1: Abreviaturas

Abreviatura y significado en español		Como se encuentran en el estado del Arte	
ACA	Algoritmo de colonia de Abejas	ABC	Algorithm Bee Colony
BIPAG	Biobjetivo PAG	BIGAP	Biobjective GAP
EMLV	Escala muy larga de vecindario	VLSN	Very large-scale neighborhood
BTRA	Búsqueda Tabú y Ramificación y acotamiento	TSBB	Tabu search and Branch and bound
SCHD	Sistema de colonia de hormigas difuso	DACS	Diffusive ant colony system
ED	Evolución diferencial	DE	Differential evolution
AL	Algoritmo de Luciérnagas	FA	Fire flies algorithm
ABG	Algoritmo de Búsqueda Gravitacional	GSA	Gravity search Algorithm
MCM	Método de composición Musical	MMC	Method of Musical Composition
IP	Inteligencia de Partículas	SI	Swarm Intelligence
VBPY	Variable de búsqueda profunda de Yagiura	VDSY	Variable Depth Search of Yagiura
BTY	Búsqueda Tabú de Yagiura	TSEC	Tabu search
BTCE	Hibrido Búsqueda Tabú y Cadena de Expulsión	TSEC	Tabu search and ejection chains
HRL	Heuristica relajación Lagrangiana	HLR	Lagrangian Relaxation Heuristic
VBP -i	Variable de búsqueda profunda técnicas de ramificación	VDS-i	Variable Depth Search-i
VBP - j	Variable de búsqueda profunda técnicas de ramificación	VDS-j	Variable Depth Search-j
BLM	Búsqueda Local Aleatoria Multicomienzo	MLS	Random multi-star local search
VBP-Y	Variable de búsqueda profunda de Yagiura *et al.*	YYI	Variable Depth Search- Yagiura *et al.*
VBP-RA	Variable de búsqueda profunda de Race y Amini .	VDS-RA	Variable Depth Search- Racer and Amini
BT-L	Búsqueda tabú de Laguna *et al.*	LKGG	Tabu search by Laguna *et al.*
BT-NI	Búsqueda tabú de Nonobe e Ibaraki	NI	Tabu Search by Nonobe and Ibaraki
AG-CB	Algoritmo Genético de Chu y Beasley	CB	Genetic Algorithm by Chu and Beasley
AGC	Algoritmo Genético constructivo	CGA	Constructive Genetic Algorithm
RRB	Replicación de rama mas baja	LBR	Lower bound replication)V Jeet, E Kutanoglu 2007
AC	Arrepentimiento codicioso	RGT	Greedy regret V Jeet, E Kutanoglu 2007
HRRBAC	Hibrido entre RRB y AC	HBR	Hibrid LBR and RGT
MRPAG	Muy larga escala de variables en búsqueda de vecindarios para el problema de asignación generalizada multi-recursos	VLSVNS	Very large-scale variable neighborhood search for the multi-resource generalized assignment problem
HA-N	Heurística de aproximación de Nauss	NAUSS	Heuristic aproac NAUSS

Cuadro A.2: Abreviaturas2

Bibliografía

[1] Albareda-Sambola, M., Van Der Vlerk, M. H., and Fernández, E. Exact solutions to a class of stochastic generalized assignment problems. *European journal of operational research 173*, 2 (2006), 465–487.

[2] Alfandari, L., Plateau, A., and Tolla, P. A two-phase path relinking algorithm for the generalized assignment problem. In *Proceedings of the Fourth Metaheuristics International Conference* (2001), pp. 175–179.

[3] Amini, M. M., and Racer, M. A rigorous computational comparison of alternative solution methods for the generalized assignment problem. *Management Science 40*, 7 (1994), 868–890.

[4] Amini, M. M., and Racer, M. A hybrid heuristic for the generalized assignment problem. *European Journal of Operational Research 87*, 2 (1995), 343–348.

[5] Asahiro, Y., Ishibashi, M., and Yamashita, M. Independentand cooperative parallel search methods for the generalized assignment problem. *Optimization methods and Software 18*, 2 (2003), 129–141.

[6] Avella, P., Boccia, M., and Vasilyev, I. A computational study of exact knapsack separation for the generalized assignment problem. *Computational Optimization and Applications 45*, 3 (2010), 543–555.

[7] Baker, B. M., and Sheasby, J. Extensions to the generalised assignment heuristic for vehicle routing. *European Journal of Operational Research 119*, 1 (1999), 147–157.

[8] Balachandar, S. R., and Kannan, K. A new heuristic approach for the large-scale generalized assignment problem. *Int J Math Comput Phys Elect Comput Eng 3* (2009), 969–974.

[9] Baykasoglu, A., Ozbakir, L., and Tapkan, P. Artificial bee colony algorithm and its application to generalized assignment problem. *Swarm Intelligence: Focus on Ant and particle swarm optimization 532* (2007), 113–144.

[10] Beasley, J. E. "OR-Library: distributing test problems by electronic mail", Journal of the Operational Research Society 41(11) (1990) pp1069-1072., 1990.

[11] Brommesson, P. Solving the generalized assignment problem by column enumeration based on lagrangian reduced costs, 2006.

[12] Cattrysse, D. G., Salomon, M., and Van Wassenhove, L. N. A set partitioning heuristic for the generalized assignment problem. *European Journal of Operational Research 72*, 1 (1994), 167–174.

[13] Chu, P. C., and Beasley, J. E. A genetic algorithm for the generalised assignment problem. *Computers & Operations Research 24*, 1 (1997), 17–23.

[14] Coello, C. A. C. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer methods in applied mechanics and engineering 191*, 11 (2002), 1245–1287.

[15] Cohen, R., Katzir, L., and Raz, D. An efficient approximation for the generalized assignment problem. *Information Processing Letters 100*, 4 (2006), 162–166.

[16] Cuevas, E., and Ortega-Sánchez, N. El algoritmo de búsqueda armónica y sus usos en el procesamiento digital de imágenes. *Computación y Sistemas 17*, 4 (2013), 543–560.

[17] De Farias, I., and Nemhauser, G. L. A family of inequalities for the generalized assignment polytope. *Operations Research Letters 29*, 2 (2001), 49–55.

[18] De Maio, A., and Roveda, C. An all zero-one algorithm for a certain class of transportation problems. *Operations Research 19*, 6 (1971), 1406–1418.

[19] Diaz, J. A., and Fernández, E. A tabu search heuristic for the generalized assignment problem. *European Journal of Operational Research 132*, 1 (2001), 22–38.

[20] Feltl, H., and Raidl, G. R. An improved hybrid genetic algorithm for the generalized assignment problem. In *Proceedings of the 2004 ACM symposium on Applied computing* (2004), ACM, pp. 990–995.

[21] Fisher, M. L., and Jaikumar, R. A generalized assignment heuristic for vehicle routing. *Networks 11*, 2 (1981), 109–124.

[22] Fisher, M. L., Jaikumar, R., and Van Wassenhove, L. N. A multiplier adjustment method for the generalized assignment problem. *Management Science 32*, 9 (1986), 1095–1103.

[23] French, A. P., and Wilson, J. M. An lp-based heuristic procedure for the generalized assignment problem with special ordered sets. *Computers & operations research 34*, 8 (2007), 2359–2369.

[24] Geem, Z. W., Kim, J. H., and Loganathan, G. V. A new heuristic optimization algorithm: harmony search. *simulation 76*, 2 (2001), 60–68.

[25] Gottlieb, E. S., and Rao, M. The generalized assignment problem: Valid inequalities and facets. *Mathematical Programming 46*, 1 (1990), 31–52.

[26] Guignard, M., and Rosenwein, M. B. Technical note—an improved dual based algorithm for the generalized assignment problem. *Operations Research 37*, 4 (1989), 658–663.

[27] Haddadi, S., and Ouzia, H. Effective algorithm and heuristic for the generalized assignment problem. *European Journal of Operational Research 153*, 1 (2004), 184–190.

[28] Hallefjord, Å., Jörnsten, K. O., and Värbrand, P. Solving large scale generalized assignment problems—an aggregation/disaggregation approach. *European Journal of Operational Research 64*, 1 (1993), 103–114.

[29] Higgins, A. J. A dynamic tabu search for large-scale generalised assignment problems. *Computers & operations research 28*, 10 (2001), 1039–1048.

[30] Jeet, V., and Kutanoglu, E. Lagrangian relaxation guided problem space search heuristics for generalized assignment problems. *European Journal of Operational Research 182*, 3 (2007), 1039–1056.

[31] Jörnsten, K., and Näsberg, M. A new lagrangian relaxation approach to the generalized assignment problem. *European Journal of Operational Research 27*, 3 (1986), 313–323.

[32] Kar, S., Basu, K., and Mukherjee, S. Solution of generalized fuzzy assignment problem with restriction on costs under fuzzy environment. *International journal of fuzzy mathematics and systems 4* (2014), 169–180.

[33] Karabakal, N., and Bean, J. A steepest descent multiplier adjustment method for the generalized assignment problem. Tech. rep., Univ. of Michigan, Ann Arbor, MI (United States), 1994.

[34] Klastorin, T. D. An effective subgradient algorithm for the generalized assignment problem. *Computers & Operations Research 6*, 3 (1979), 155–164.

[35] Krumke, S. O., and Thielen, C. The generalized assignment problem with minimum quantities. *European Journal of Operational Research 228*, 1 (2013), 46–55.

[36] Laguna, M., Kelly, J. P., González-Velarde, J., and Glover, F. Tabu search for the multilevel generalized assignment problem. *European journal of operational research 82*, 1 (1995), 176–189.

[37] Litvinchev, I., Mata, M., Rangel, S., and Saucedo, J. Lagrangian heuristic for a class of the generalized assignment problems. *Computers & Mathematics with Applications 60*, 4 (2010), 1115–1123.

[38] Liu, L., Mu, H., Song, Y., Luo, H., Li, X., and Wu, F. The equilibrium generalized assignment problem and genetic algorithm. *Applied Mathematics and Computation 218*, 11 (2012), 6526–6535.

[39] Liu, Y. Y., and Wang, S. A scalable parallel genetic algorithm for the generalized assignment problem. *Parallel computing 46* (2015), 98–119.

[40] Lorena, L. A., Narciso, M. G., and Beasley, J. A constructive genetic algorithm for the generalized assignment problem. *Evolutionary Optimization 5* (2002), 1–19.

[41] Lorena, L. A. N., and Narciso, M. G. Relaxation heuristics for a generalized assignment problem. *European Journal of Operational Research 91*, 3 (1996), 600–610.

[42] Lourenço, H. R. D., de la Figuera, D. S., and BBVA, F. *Métodos de solución de problemas de asignación de recursos sanitarios*. Fundación BBVA, 2004.

[43] Martello, S., and Toth, P. A bound and bound algorithm for the zero-one multiple knapsack problem. *Discrete Applied Mathematics 3*, 4 (1981), 275–288.

[44] Mazzola, J. B. Generalized assignment with nonlinear capacity interaction. *Management Science 35*, 8 (1989), 923–941.

[45] Mitrović-Minić, S., and Punnen, A. P. Local search intensified: Very large-scale variable neighborhood search for the multi-resource generalized assignment problem. *Discrete Optimization 6*, 4 (2009), 370–377.

[46] Moccia, L., Cordeau, J.-F., Monaco, M. F., and Sammarra, M. A column generation heuristic for a dynamic generalized assignment problem. *Computers & Operations Research 36*, 9 (2009), 2670–2681.

[47] Mora-Gutiérrez, R. A., Ramírez-Rodríguez, J., Rincón-García, E. A., Ponsich, A., and Herrera, O. An optimization algorithm inspired by social creativity systems. *Computing 94*, 11 (2012), 887–914.

[48] Munapo, E., Lesaoana, M., Nyamugure, P., and Kumar, S. A transportation branch and bound algorithm for solving the generalized assignment problem. *International Journal of System Assurance Engineering and Management 6*, 3 (2015), 217–223.

[49] Nauss, R. M. Solving the generalized assignment problem: An optimizing and heuristic approach. *INFORMS Journal on Computing 15*, 3 (2003), 249–266.

[50] Nutov, Z., Beniaminy, I., and Yuster, R. A (1–1/e)-approximation algorithm for the generalized assignment problem. *Operations Research Letters 34*, 3 (2006), 283–288.

[51] Ocampo, E. M. T., and Granada, M. Método híbrido entre el algoritmo genético de chu-beasley y simulated annealing para la solución del problema de asignación generalizada. *Scientia et technica 2*, 28 (2005).

[52] Osman, I. H. Heuristics for the generalised assignment problem: simulated annealing and tabu search approaches. *Operations-Research-Spektrum 17*, 4 (1995), 211–225.

[53] Pigatti, A., De Aragao, M. P., and Uchoa, E. Stabilized branch-and-cut-and-price for the generalized assignment problem. *Electronic Notes in Discrete Mathematics 19* (2005), 389–395.

[54] Posta, M., Ferland, J. A., and Michelon, P. An exact method with variable fixing for solving the generalized assignment problem. *Computational Optimization and Applications 52*, 3 (2012), 629–644.

[55] Racer, M., and Amini, M. M. A robust heuristic for the generalized assignment problem. *Annals of Operations Research 50*, 1 (1994), 487–503.

[56] Rainwater, C., Geunes, J., and Romeijn, H. E. The generalized assignment problem with flexible jobs. *Discrete Applied Mathematics 157*, 1 (2009), 49–67.

[57] Rajabi-Alni, F. Two exact algorithms for the generalized assignment problem. *arXiv preprint arXiv:1303.4031* (2013).

[58] Ramalhinho-Lourenço, H., and Serra, D. Heurísticas adaptativas para el problema de asignación generalizada.

[59] Rashedi, E., Nezamabadi-Pour, H., and Saryazdi, S. Gsa: a gravitational search algorithm. *Information sciences 179*, 13 (2009), 2232–2248.

[60] Romeijn, H. E., and Romero Morales, D. Generating experimental data for the generalized assignment problem. *Operations Research 49*, 6 (2001), 866–878.

[61] Ross, G. T., and Soland, R. M. A branch and bound algorithm for the generalized assignment problem. *Mathematical programming 8*, 1 (1975), 91–103.

[62] Ross, G. T., and Soland, R. M. Modeling facility location problems as generalized assignment problems. *Management Science 24*, 3 (1977), 345–357.

[63] Sahni, S., and Gonzalez, T. P-complete approximation problems. *Journal of the ACM (JACM) 23*, 3 (1976), 555–565.

[64] Sahu, A., and Tapadar, R. Solving the assignment problem using genetic algorithm and simulated annealing. In *IMECS* (2006), pp. 762–765.

[65] Savelsbergh, M. A branch-and-price algorithm for the generalized assignment problem. *Operations research 45*, 6 (1997), 831–841.

[66] Sethanan, K., and Pitakaso, R. Improved differential evolution algorithms for solving generalized assignment problem. *Expert Systems with Applications 45* (2016), 450–459.

[67] Shmoys, D. B., and Tardos, É. An approximation algorithm for the generalized assignment problem. *Mathematical programming 62*, 1-3 (1993), 461–474.

[68] Shtubt, A. Modelling group technology cell formation as a generalized assignment problem. *THE INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH 27*, 5 (1989), 775–782.

[69] Stucchi, L. Optimización de un problema de asignación generalizada a partir de un algoritmo de colonia de hormigas que incorpora un mecanismo de difusión.

[70] Tapkan, P., ÖZbakiR, L., and BaykasoğLu, A. Solving fuzzy multiple objective generalized assignment problems directly via bees algorithm and fuzzy ranking. *Expert systems with applications 40*, 3 (2013), 892–898.

[71] Tasgetiren, M. F., Suganthan, P. N., Chua, T. J., and Al-Hajri, A. Differential evolution algorithms for the generalized assignment problem. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on* (2009), IEEE, pp. 2606–2613.

[72] Trick, M. A. A linear relaxation heuristic for the generalized assignment problem. *Naval Research Logistics (NRL) 39*, 2 (1992), 137–151.

[73] Woodcock, A. J., and Wilson, J. M. A hybrid tabu search/branch & bound approach to solving the generalized assignment problem. *European journal of operational research 207*, 2 (2010), 566–578.

[74] Yagiura, M., Ibaraki, T., and Glover, F. An ejection chain approach for the generalized assignment problem. *INFORMS Journal on Computing 16*, 2 (2004), 133–151.

[75] Yagiura, M., Ibaraki, T., and Glover, F. A path relinking approach with ejection chains for the generalized assignment problem. *European journal of operational research 169*, 2 (2006), 548–569.

[76] Yagiura, M., Yamaguchi, T., and Ibaraki, T. A variable depth search algorithm with branching search for the generalized assignment problem. *Optimization Methods and Software 10*, 2 (1998), 419–441.

[77] Yang, X.-S. *Nature-inspired metaheuristic algorithms*. Luniver press, 2010.

[78] Zhang, C. W., and Ong, H. L. An efficient solution to biobjective generalized assignment problem. *Advances in Engineering Software 38*, 1 (2007), 50–58.